



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA FOOD SERVICE
UTILIZANDO A PLATAFORMA ANDROID

Araranguá

2018

DEIVID JONATAN DA ROSA
EVERTON NIOTTI FAVARO

DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA FOOD SERVICE
UTILIZANDO A PLATAFORMA ANDROID

Trabalho Conclusão do Curso de Graduação em
Tecnologias da Informação e Comunicação do
Centro de Araranguá da Universidade Federal de
Santa Catarina como requisito para a obtenção do
Título de Bacharel em Tecnologias da Informação
e Comunicação. Orientador: Prof. Dr. Robson
Rodrigues Lemos

Araranguá
2018

Rosa, Deivid Jonatan da,

Desenvolvimento de um aplicativo móvel para food service utilizando a plataforma Android / Deivid Jonatan da Rosa ; Everton Niotti Favaro ; orientador Dr. Robson Rodrigues Lemos , 2018.

101 p.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Tecnologias da Informação e Comunicação, Araranguá, 2018.

Inclui referências.

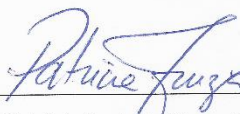
1. Tecnologias da Informação e Comunicação. I. Favaro, Everton
Everton Niotti. II. Lemos, Robson Rodrigues. III. Universidade Federal de Santa Catarina. Graduação em Tecnologias da Informação e Comunicação. IV. Título.

Ficha catalográfica elaborada por Débora Maria Russiano Pereira, CRB-14/1125

DEIVID JONATAN DA ROSA
EVERTON NIOTTI FAVARO

DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA FOOD SERVICE
UTILIZANDO A PLATAFORMA ANDROID

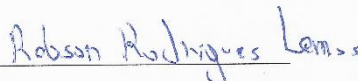
Este Trabalho de Conclusão de Curso foi julgado adequado para a
Obtenção do Título de Bacharel em Tecnologias da Informação e
Comunicação, e aprovado em sua forma final pelo Curso de Graduação
Em Tecnologias da Informação e Comunicação.



Prof. Patricia Jantsch Fiuza, Dr^a.


Coordenador do Curso

Banca Examinadora:



Prof. Dr. Robson Rodrigues Lemos

Orientador



Prof. Helio Aisenberg Ferenhof, Dr.

Universidade Federal de Santa Catarina



Prof. Patricia Jantsch Fiuza, Dr^a.

Universidade Federal de Santa Catarina

Dedicamos este trabalho às pessoas com quem convivemos nesses espaços ao longo desses anos. As experiências de uma produção compartilhada na comunhão com amigos nesses espaços foram a melhores experiências da nossa formação acadêmica. A todos aqueles que de alguma forma estiveram e estão próximos de nós, fazendo a vida valer cada vez mais a pena.

AGRADECIMENTOS

Agradecer primeiramente a Deus, que me deu energia e sabedorias para que conseguir completar mais essa etapa de minha vida.

Agradecer também a minha família, minha esposa Jamili Barbosa pelo carinho e compreensão que me deu ao decorrer do curso, minha filha que me deu um foco na minha vida e me deu força para aguentar os momentos mais difíceis.

Agradeço também minha mãe e que me deu incentivou em todos os momentos que estive na faculdade, me dando força e motivação para continuar.

Aos colegas que no decorrer dessa caminhada foram de extrema importância para minha formação compartilhando conhecimento e me ensinando o espírito de equipe.

Ao professor e Orientador DR. Robson Rodrigues Lemos por sua paciência nas orientações e sua dedicação em todos os momentos, onde tornou possível a conclusão desse trabalho.

Agradeço a todos os professores que proporcionaram experiências e conhecimentos únicos, que será lembrado ao decorrer da minha vida.

“Deivid Jonatan da Rosa”

Agradecer primeiramente a Deus, que me deu esse desafio. A todos os meus amigos e familiares que me apoiaram.

Agradeço a todos os meus colegas que conheci nessa caminhada e a todos que tomaram uma cerveja comigo durante esses 5 anos.

Agradeço a minha amiga Joana que me apoiou e me ajudou nessa caminhada e todos os amigos que aguentaram minhas reclamações.

Ao professor e Orientador DR. Robson Rodrigues Lemos por sua paciência nas orientações e sua dedicação em todos os momentos, onde tornou possível a conclusão desse trabalho.

“Everton Niotti Favaro”

“A melhor maneira de prever o futuro é criá-lo” Peter
Drucker

RESUMO

Este trabalho de conclusão de curso, tem como objetivo a elaboração de um projeto e desenvolvimento de uma aplicação para FoodService em Android, para a empresa Tia Graça salgadinhos. A partir de uma entrevista com a proprietária foram constatadas dificuldades no armazenamento dos dados dos pedidos realizados pelos clientes. A partir disso, foi elaborado um projeto de aplicativo utilizando a ferramenta de desenvolvimento Android Studio com o objetivo de facilitar a organização da empresa em relação aos pedidos de compra de comida por parte dos clientes. Após os testes iniciais com o aplicativo, a proprietária relatou que o aplicativo facilitará o modo de agendar suas encomendas e em função disso será possível aumentar a sua produtividade e diminuir as inconsistências regulares no sistema de agendamento de pedidos atual. Durante o teste do aplicativo com alguns clientes, obteve-se um resultado positivo, devido ao fato de facilitar a realização dos pedidos e fornece um controle maior para o acompanhamento do mesmo.

Palavras-chave: Android; FoodService; Aplicativo.

ABSTRACT

This course completion work aims to design and develop an application for FoodService on Android, for the company Tia Graça salgadinhos. From an interview with the owner, there were difficulties in storing the data of the orders made by the customers. From this, an application project was developed using the Android Studio development tool to facilitate the organization of the company in relation to customers' purchase requests for food. After initial testing with the application, the owner reported that the application will make it easier for them to schedule their orders and increase their productivity and reduce the regular inconsistencies in the current order-scheduling system. During the test of the application with some clients, a positive result was obtained, due to the fact to facilitate the accomplishment of the requests and to provide a greater control for the follow-up of the same.

Keywords: Android; FoodService; App.

LISTA DE FIGURAS

Figura 1 - Locais onde os Brasileiros fazem refeições fora do lar.....	22
Figura 2 - Pesquisa de canais de atendimento segundo IFB.....	23
Figura 3 - informações da estimativa de faturamento do ifood no decorrer dos até o ano de 2017.....	30
Figura 4 - Imagem do ifood em vários canais de vendas na internet	31
Figura 5 - Telas do Aplicativo do Burger King	32
Figura 6 - Telas do aplicativo desenvolvido pela ALFALABS para empresa <i>Foodservices</i>	33
Figura 7 - Gráfico de utilização de versões do Android.....	36
Figura 8 - Android Studio Estrutura de arquivos.....	39
Figura 9 - Imagens de como e compilado um código em Java.	40
Figura 10 - Diagrama de funcionamento de um <i>Web Service</i> utilizando REST.....	42
Figura 11 - Diagrama de funcionamento de uma <i>web Service</i> utilizando SOAP	43
Figura 12 - Cantina da Tia Graça 2009	48
Figura 13 - Imagem da Cantina da Tia Graça Restaurante ano 2018	49
Figura 14 - Agenda de pedido Tia Graça Salgadinhos.....	51
Figura 15 - Fluxograma de funcionamento da solução proposta.....	52
Figura 16 - Tela de <i>Login</i>	53
Figura 17 - Tela de Cadastro	54
Figura 18 - Tela Menu App Tia Graça Salgadinho	55
Figura 19 - Tela calcular App Tia Graça Salgadinho.....	56
Figura 20 - Tela selecionar mini-salgadinhos e tela selecionar salgados grandes do App Tia Graça Salgadinho	57
Figura 21 - Tela quantidade mini-salgadinhos e tela quantidade salgados grandes .	58
Figura 22 - Tela para conferir os dados dos pedidos e informar a forma de pagamento do App Tia Graça Salgadinho	59
Figura 23 - Tela acompanha pedido do App Tia Graça Salgadinho	60
Figura 24 - Tela acompanha pedido do App Tia Graça Salgadinho	61
Figura 25 - Tela para aceitar pedidos Desktop	62
Figura 26 - Tela inicial administrador no aplicativo	63
Figura 27 - Tela de para exibir pedidos.....	64
Figura 28 - Modelo lógica do banco de dados.....	66
Figura 29 - Implementação conexão.php	67

Figura 30 - Implementação do arquivo <code>logar.php</code>	67
Figura 31 - Implementação do Arquivo <code>cadastrar.php</code>	68
Figura 32 - Implementação do arquivo <code>pedidocad.php</code> parte 1	68
Figura 33 - Implementação do arquivo <code>pedidocad.php</code> parte 2	69
Figura 34 - Implementação do arquivo <code>AcompanharPedido.php</code>	69
Figura 35 - Implementação do arquivo <code>RecuperarDadosUsuario.php</code>	70
Figura 36 - Implementação do arquivo <code>pedidoinfo.php</code>	70
Figura 37 - Implementação do arquivo <code>RetonarDadosPedido.php</code>	70
Figura 38 - Implementação do arquivo <code>RetornarInfoPedidos.php</code>	71
Figura 39 - Implementação do arquivo <code>recuperarID.php</code>	71
Figura 40 - Implementação do arquivo <code>AceitarPedido.php</code>	71
Figura 41 - Implementação do arquivo <code>RecusarPedido.php</code>	72
Figura 42 - Implementação do arquivo <code>PedidosAguardandoConfirmacao.php</code>	72
Figura 43 - Implementação do arquivo <code>RetornarTodososPedidoDoDia.php</code>	73
Figura 44 - Implementação do arquivo <code>RetornarTodososPedidos.php</code>	73
Figura 45 - Estrutura do aplicativo no Android Studio	74
Figura 46 - Método <code>logar</code> da <code>MainActivity.java</code>	75
Figura 47 - Métodos <code>PedidoSalgadinhos()</code> e <code>PedidosSalgados()</code> da aplicação	75
Figura 48 - <i>ListView</i> preenchimento	76
Figura 49 - <i>ListView adapter</i>	76
Figura 50 - <i>ListView</i> visualização	77
Figura 51 - <i>ListView</i> visualização	77
Figura 52 - Função <code>pegar dados</code> do <i>EditText</i>	78
Figura 53 - <i>ListView</i> com <i>EditText</i>	78
Figura 54 - Implementação do cadastro do pedido	79
Figura 55 - Implementação da ação do botão para aceitar pedido	79
Figura 56 - Implementação da ação do botão para recusar pedido	80

LISTA DE ABREVIATURAS E SIGLAS

ABF-Associação Brasileira de Franchising

ABNT -Associação Brasileira de Normas Técnicas

ABRASEL-Associação Brasileira de Bares e Restaurantes

AMESC - Associação dos Municípios do Extremo Sul Catarinense

API - Application Programming Interface

APP – Application

B2A - Business to administration

B2B - Business to business

B2C - Business to consumer

B2E - Business to employee

C2A - Consumer to administration

C2C - Consumer to consumer

CPF - Contribuinte Pessoa Física

CSS - Cascading Style Sheets

HTTP - Hyper Text Transfer Protocol

IBGE – Instituto Brasileiro de Geografia e Estatística

IDE - Integrated Development Environment

IFB - Instituto FoodService Brasil

IP - Internet Protocol

ONU - Organização das nações Unidas

PHP - Personal Home Page

Pnad C - Pesquisa Nacional por Amostragem de Domicílios Contínua

REST - Representational State Transfer

SOAP - Simple Object Data Protocol

SGBD - Sistema Gerenciador de Banco de Dados

TCP - Transmission Control Protocol

TIC – Tecnologia da Informação e Comunicação

UI - Users Interface

URL - Uniform Resource Locator

WEB - World Wide Web

XML - Extensible Markup Language

Sumário

SUMÁRIO.....	1
1 INTRODUÇÃO.....	15
1.1 PROBLEMÁTICA.....	16
1.2 OBJETIVOS	17
1.2.1 Objetivo Geral.....	17
1.2.2 Objetivos Específicos.....	17
1.3 JUSTIFICATIVA.....	17
1.4 METODOLOGIA	19
1.5 ORGANIZAÇÃO DO TRABALHO	20
2 FUNDAMENTAÇÃO TEÓRICA	21
2.1 ALIMENTAÇÃO FORA DO LAR (AFL)	21
2.1.1 Razões para o crescimento do <i>FoodService</i>	22
2.2 PRINCIPAIS CANAIS DE ATENDIMENTO OFERECIDOS EM EMPRESAS <i>FOODSERVICE</i>	22
2.2.1 Serviço de atendimento <i>On Premise</i>	23
2.2.2 Serviço de atendimento <i>Take away</i>	24
2.2.3 Serviço de atendimento via telefone e <i>delivery</i>	24
2.2.4 Pedidos via internet	24
2.3 INTERNET	25
2.3.1 Negócios na Internet.....	25
2.3.2 Comércio eletrônico (<i>e-commerce</i>).....	26
2.3.3 <i>Business to Consumer (B2C)</i>	28
2.4 SOLUÇÕES B2C PARA <i>FOODSERVICES</i>	28
2.5 APLICAÇÕES MÓVEIS DE EMPRESA <i>FOODSERVICE</i>	29
2.5.1 Ifood	29
2.5.2 Burger King.....	31
2.5.3 Alpha labs	32
3 ESTUDOS DAS TECNOLOGIAS PARA O DESENVOLVIMENTO DA SOLUÇÃO EM ESTUDO	35
3.1 SISTEMA ANDROID.....	35
3.2 <i>INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)</i>	37
3.2.1 Desenvolvendo aplicativos para Google usando o sistema Android	37

3.2.2	Android SDK (<i>Software development kit</i>)	38
3.2.3	IDE Android Studio	38
3.2.4	<i>Manisfest</i> -Android Manifest	39
3.2.5	Módulo de classes JAVA	40
3.2.6	Recursos Layout XML Android	41
3.3	<i>WEB SERVICES</i>	41
3.3.1	Formas de transmissão de dados <i>Web Services</i>	42
3.4	SERVIDORES DE BANCOS DE DADOS	43
3.5	MYSQL (SGBD)	44
3.6	A LINGUAGEM PHP	44
3.7	A LINGUAGEM DE MARCAÇÃO HTML (<i>HIPERTEXT MARKUP LANGUAGE</i>)	45
3.8	JAVA SCRIPT	45
3.9	<i>CASCADING STYLE SHEETS</i>	46
4	PROJETO DO APLICATIVO DE <i>FOODSERVICE</i>	47
4.1	CANTINA DA TIA GRAÇA E TIA GRAÇA SALGADINHO	47
4.2	SITUAÇÃO ATUAL DA CANTINA DA TIA GRAÇA RESTAURANTE E TIA GRAÇA SALGADINHOS	49
4.3	PROBLEMAS ENCONTRADOS NA GESTÃO DE PEDIDOS NA CANTINA TIA GRAÇA SALGADINHOS	50
4.4	PROPOSTA DE UM SISTEMA DE AGENDAMENTO DE PEDIDOS NA TIA GRAÇA SALGADINHOS	51
4.5	PROPOSTA DAS TELAS DE NAVEGAÇÃO DO APLICATIVO	52
4.5.1	Tela de <i>login</i>	53
4.5.2	Tela cadastro	53
4.5.3	Tela de menu	54
4.5.4	Tela de cálculo	55
4.5.5	Tela de seleção de itens	56
4.5.6	Tela de quantidades de itens selecionados	57
4.5.7	Tela de informações do pedido e formas de pagamento	58
4.5.8	Tela de acompanhamento do pedido	59
4.5.9	Tela de acompanhamento do pedido e informação	60
4.6	PROPOSTA DE TELAS PARA CONFIRMAÇÃO DE PEDIDOS	61
4.6.1	Proposta para desktop	62

4.6.2	Proposta para mobile	63
5	DESENVOLVIMENTO DO APLICATIVO	65
5.1	CLIENTE-SERVIDOR	65
5.2	BANCO DE DADOS	65
5.3	A LINGUAGEM PHP.....	66
5.4	IMPLEMENTAÇÃO EM PHP.....	66
5.5	ANDROID STUDIO.....	73
5.5.1	Funcionalidades da aplicação	73
6	TESTE DO APLICATIVO DESENVOLVIDO NO AMBIENTE DE TRABALHO	81
6.1	PERGUNTAS DISSERTATIVAS APRESENTADAS PARA OS USUÁRIOS	81
6.2	ANÁLISES DAS RESPOSTAS DOS CLIENTES	81
6.3	ANÁLISE DAS RESPOSTAS DOS FUNCIONÁRIOS DA EMPRESA.....	82
6.4	COMENTÁRIOS FINAIS.....	83
7	CONCLUSÃO	85
7.1	TRABALHOS FUTUROS	85
8	REFERÊNCIAS	87
9	ANEXOS.....	93

1 INTRODUÇÃO

No decorrer dos anos a expansão das Tecnologias da Informação e Comunicação (TICs) estão cada vez mais presentes no nosso cotidiano, um exemplo disso é que em 2016 o Brasil fechou o ano, com 116 milhões de pessoas com acesso à internet, sendo esse número equivalente de 64,7 % da população. Informação obtida por meio da Pesquisa Nacional por Amostragem de Domicílios contínua (Pnad C), e divulgada pelo Instituto Brasileiro de Geografia e Estatística (IBGE) [...]. (G1,2018).

Junto a essa expansão tecnológica, outros setores como o da alimentação tiveram um aumento nos últimos anos. No setor da alimentação espera dobrar seu percentual em 2018, no ano de 2017 seu crescimento foi de 2,5%. Em 2018 tem-se como uma estimativa um crescimento de 4,5%, seu aquecimento se dá pelas novas leis trabalhista e redução de confrontos de empregados e empregadores. (ABRASEL,2017)

No setor de alimentação as empresas que são caracterizadas *FoodService*, empresas que tem seu objetivo principal a produção de alimentos. Elas focam diretamente no consumidor final, englobam a produção, distribuição de produtos, fabricante de matéria prima, fornecedores de equipamentos e serviços, e também as empresas que preparam e fornecem refeições e seus transportes. Os principais segmentos do *FoodService* são: *fast foods*, restaurantes, padarias e confeitarias, hotéis e motéis, refeições coletivas, *catering* e *vending machines*. (ACOM SISTEMAS, 2014).

Segundo Content (2017), a partir de dados da Associação Brasileira de Franchising (ABF), o segmento de alimentação abriga a maior parte das empresas de *FoodService*, seu faturamento atingiu cerca 9,935 bilhões de reais em 2017. Com um aumento de 6% comparando com o ano anterior. Mantendo o hábito de se alimentar fora do lar, mesmo havendo uma crise econômica, o segmento inova rapidamente, para atender tendências e públicos exigentes.

Acompanhando esses avanços tecnológicos, com clientes cada vez mais exigentes e possuem as informações ao seu alcance, na palma de sua mão. Os clientes buscam uma interação maior com as empresas e tem novas maneiras de obter os serviço. Dentro deste contexto, uma nova maneira de interagir com o cliente é por meio de telefones celulares, *tablets* e computadores, ou seja, dispositivos que possuem conexão com à internet. (SILVA, 2015).

Segundo Paulo Solmucci, presidente executivo da Abrasel, a tecnologia vem impactando cada vez mais no setor, segundo ele, os aplicativos móveis (APPs), já estão em

torno de 40% das empresas do ramo, entregando algum tipo de serviço ou produtos podendo fazer o agendamento de pedidos ou reservas, caracterizando essa tecnologia como um avanço no setor e para os clientes dando-os maiores opções e comodidades. (ABRASEL, 2017).

As empresas *FoodServices* estão migrando da forma tradicional para uma forma de atendimento usando aplicações móveis para fornecer um melhor serviço, em um ambiente virtual, diferenciando do padrão tradicional onde o atendimento é feito por pessoas com costumes e características de atendimentos diferentes umas das outras.

A proposta deste trabalho de conclusão de curso é analisar os serviços de atendimento de empresas *FoodService*, além disso será apresentado o projeto de desenvolvimento de uma aplicação móvel para a empresa Tia Graça Salgadinhos que atua no setor produção de alimentos para festas, na cidade de Sombrio, pertencente a região AMESC (Associação dos Municípios do Extremo Sul Catarinense) no estado de Santa Catarina.

1.1 PROBLEMÁTICA

Observando os novos hábitos de clientes que buscam se alimentar fora do lar, procurando empresas *FoodServices* para fazer sua refeição, observa-se que estas empresas estão se adaptando as novas tecnologias existente para melhor atender seu cliente em potencial.

Nas últimas décadas surgiram novas rotinas para a população, tem-se pouco tempo para preparar a própria refeição, com horários de almoço cada vez menores, justificados pela demanda de trabalho das empresas empregadoras, além de que o deslocamento para o trabalho torna-se cada vez mais longo. (SCUADRA, 2018).

A partir da necessidade de optar por refeições rápidas e práticas, a empresa *FoodServices* oferece esse tipo de alimentação rápida, fator diferencial para o crescimento das empresas, a praticidade oferecida em novos canais de vendas proporcionando novas experiências de consumo tornando mais prático e rápido a hora de fazer o pedido.

Não há mais a necessidade de procurar nos catálogos telefônicos para localizar o número ou o endereço das empresas, com o avanço da tecnologia, são utilizadas aplicações móveis para fazer o pedido ou o agendamento do pedido para seus clientes. (FONTANA JUNIOR, 2013).

Quando o atendimento é feito via ligação telefônica, toda a informação da empresa tem que ser passada em uma chamada por voz, caso o cliente queira pedir uma pizza o atendente teria de falar todos os tipos de pizzas que a empresa dispõe em seu cardápio, além de informações essenciais como o preço de cada tamanho, formas de pagamento, e o tipo de

serviços para entrega. O que torna o atendimento demorado, e por vezes ocasiona problemas de comunicação por limitações no serviço telefônico gerando desconfortos para clientes que tem pouco tempo para fazer o pedido. (FONTANA JUNIOR, 2013).

1.2 OBJETIVOS

Nessa seção é apresentado o objetivo geral e objetivos específicos do projeto em questão.

1.2.1 Objetivo Geral

Projetar e desenvolver um aplicativo para solucionar os problemas analisados na forma de agendar pedidos.

1.2.2 Objetivos Específicos

Os objetivos específicos do trabalho são os seguintes:

- Explicitar empresas *FoodService* e pesquisa de soluções existentes para o agendamento de pedidos em seu canal de venda;
- Elicitar requisitos para o desenvolvimento do sistema para a empresa em estudo;
- Estudar tecnologias para o desenvolvimento de aplicativos móveis, assim como suas principais características de desenvolvimento;
- Projetar um aplicativo móvel para o agendamento de pedidos;
- Desenvolver a solução proposta para empresa, e um aplicativo móvel;
- Implantar o aplicativo móvel na empresa da região.

1.3 JUSTIFICATIVA

O segmento de *FoodService* tem um potencial enorme de expansão, com as mudanças de hábitos dos consumidores e surgem novas plataformas e aplicativos com o objetivo de

comercializar produtos e serviços. O que permite a possibilidade de interação de ambientes físicos e virtuais para as empresas *FoodService* utilizem serviços de tecnologias móveis. (CONTENT, 2017).

Segundo Luciana Lessa analista de negócio do Sebrae-MG, os serviços virtuais oferecidos por empresas vieram para ficar. (ABRASEL, 2018). Prezando a importância dos *smartphones* no cotidiano da população, as aplicações próprias das empresas, são tendências hoje para esse segmento, oferecendo para seu cliente um acesso fácil e rápido aliando comodidade para serviço, não esquecendo também de aplicações onde temos várias empresas *FoodServices* em um único local que são chamadas de *Marketplaces*.

Para Sampaio (2018) *Marketplace* é uma vitrine em um shopping center virtual, a mesma coisa que uma loja física, onde as pessoas podem entrar e comprar produtos de outra loja em um ambiente virtual, com várias lojas dos mesmos seguimentos, em que podem escolher a loja e o produto que mais lhe agradar.

Segundo um levantamento do aplicativo de pedidos de comida *ifood*, publicado no site do G1, hoje as empresas estão buscando se adaptar ao mercado, para atender melhor seu cliente tem de buscar diferenciais tecnológicos para melhorar o atendimento e aumentar faturamento da empresa. A partir do uso do aplicativo levantou-se que na região de Campinas em São Paulo no ano de 2014 para 2015 o serviço de pedidos de comida teve um aumento de 300% de pedidos mensais para a empresa depois do uso do aplicativo para fazer seus pedidos, diz Andrei Zanalieto, coordenador comercial do *ifood* (G1,2016).

Para se adaptar ao setor busca-se inovar e otimizar os serviços por meio da tecnologia e inovação a seu favor, melhorar todo o processo como controles de caixa, evitar perdas e despesas desnecessárias, e também conhecer melhor seu cliente é um dos grandes desafios dessas empresas. A partir da criação de programas que fidelizem os seus clientes em diferentes canais de atendimento e cardápios digitais em que é possível registrar informações e costumes de seus clientes. (CONTENT, 2017).

O projeto proposto tem como foco a empresa da mãe de um dos autores, que devido a acessibilidades das informações, se tornou a melhor opção, esta empresa é a Tia Graça Salgadinhos, que se caracteriza como uma empresa do ramo *FoodService*, onde a aplicação será um serviço *web*, com o objetivo de facilitar o agendamento de pedidos da empresa, gerar um novo canal de venda para atender melhor os seus clientes. Esse canal de venda não vai ser único exclusivo para atendimento, a empresa continuará a atender os seus serviços de agendamento também por telefone e mensagens.

1.4 METODOLOGIA

A primeira etapa desse trabalho se inicia com um estudo sobre empresas que atuam no ramo *FoodService*, identificando suas características para serem denominadas nesse setor, buscando seus problemas e as suas necessidades, além de estudar algumas soluções para aplicações que utilizam serviço de atendimento de pedidos para esse setor.

Para fundamentar o trabalho foi feita uma pesquisa bibliográfica exploratória para amparar o estudo, analisar trabalhos já existentes sobre *FoodService* e o ramo alimentício, e identificar soluções móveis para empresas que são do mesmo segmento ou semelhantes.

Pesquisa bibliográfica: quando elaborada de material já publicado, e constitui principalmente de livros, revistas, publicações em periódicos e artigos científicos, jornais, boletins, monografias, dissertações, teses etc. onde tem como objetivo colocar o pesquisador em contato direto com todo o material já escrito sobre o assunto de pesquisa. (PRODANOV; FREITAS, 2013, p.54).

Como aplicação tecnológica prática em nosso estudo, analisamos a rotina diária da empresa localizada em Sombrio em Santa Catarina, Tia Graça salgadinhos, com mais de 10 anos de funcionamento, na data de 20 março de 2018. A equipe fez a primeira visita a empresa com o intuito de analisar como a empresa realiza o agendamento de seus pedidos e quais são os principais canais de atendimento da empresa, por meio de uma entrevista informal com dona da empresa e seus funcionários, e por meio de fotos do atual sistema de agendamento, para que fosse possível identificar os problemas e encontrar soluções. Na segunda visita feita na empresa dia 27 de outubro de 2018, o grupo apresentou a solução proposta, um aplicativo desenvolvido em Android, para a dona e os funcionários da empresa e para um grupo de 7 clientes, e foi pedido para que eles realizassem o teste do aplicativo e relatassem como foi a experiência.

Para o desenvolvimento do aplicativo utilizou-se tecnologias móveis para a plataforma Android.

Para a parte do *web service* utilizou-se do banco de dados MYSQL, e de um servidor *web*, onde ficou hospedados os arquivos em PHP e o banco de dados.

No projeto de interface do aplicativo seguiu-se as etapas de projeto do ciclo de engenharia de usabilidade.

E, para avaliação do aplicativo adotou-se questões descritas a partir de entrevistas com clientes e funcionários da empresa.

1.5 ORGANIZAÇÃO DO TRABALHO

Esse trabalho será dividido em 6 capítulos. No primeiro capítulo é apresentado o projeto, contendo uma breve contextualização e apresentação da problemática do assunto seguindo com os objetivos gerais e específicos, justificativa e a metodologia.

No segundo capítulo se dão os estudos bibliográficos do ramo alimentício e também o estudo das soluções mais utilizadas hoje para fazer pedido de comidas detalhando aplicações móveis, seguindo o conceito *WebService*, onde as aplicações e uma interação de duas plataformas diferentes.

O terceiro capítulo deslumbra das tecnologias que serão utilizadas para esse projeto detalhamento de todos os passos para desenvolver solução para empresa, os softwares utilizados para desenvolver os layouts de tela e também o as ferramentas para programação o banco de dados utilizados.

Já, no quarto capítulo propõe o projeto do aplicativo móvel para empresa de base para esse estudo e projeto o de interface do aplicativo.

O quinto capítulo corresponde a descrição do teste piloto do aplicativo com os usuários. E, por fim, o sexto capítulo é feito a conclusão do estudo.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo apresenta-se informações do ramo de alimentação focando no tipo de alimentação fora do lar (AFL), tipos de serviços que são utilizados hoje em dia para fazer agendamento de pedidos, utilização da internet para soluções em seu cotidiano, *WebService* e aplicativos para agendamento de comidas.

2.1 ALIMENTAÇÃO FORA DO LAR (AFL)

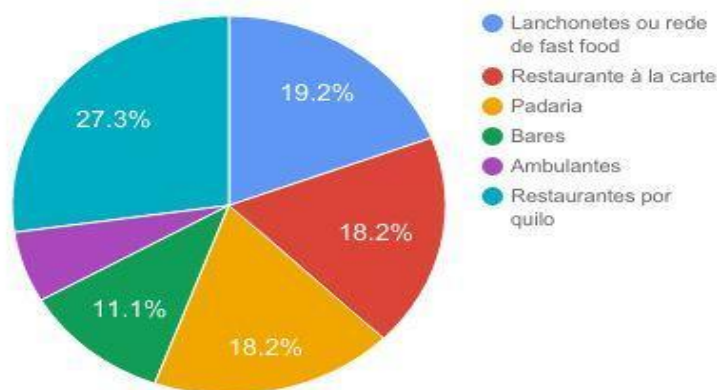
Em novos tempos, decorrentes de mudanças de consumo da sociedade, vem ocorrendo mudanças de hábitos alimentares, sendo que a tendência que vem se destacando ganhando mais força é a alimentação fora do lar ou *FoodService*. Este tipo de alimentação é constituído por restaurantes, lanchonetes, bares e outros serviços, os quais vem crescendo exponencialmente todos os anos, exigindo inovações das empresas desse segmento para estar sempre em destaque no mercado a inovação em seu serviço.

Uma pesquisa feita pela Abrasel, apontou que alimentação fora do lar fechou 2016 com um faturamento R\$154 bilhões. No ano passado bares e restaurantes tiveram um crescimento nominal de 3,47%. (MACHADO, 2017).

Como pode se observar na Figura 1, os locais onde os usuários procuram para fazer refeição fora do lar são: lanchonetes ou rede *fast food* 19.2%, restaurantes a lá carte 18.2%, padaria 18.2%, bares 11.1%, ambulantes 6% e restaurantes por quilo 27,3%.

Figura 1- Locais onde os Brasileiros fazem refeições fora do lar

Locais onde os brasileiros fazem refeições fora do lar



Fonte: (MACHADO, 2017)

2.1.1 Razões para o crescimento do *FoodService*

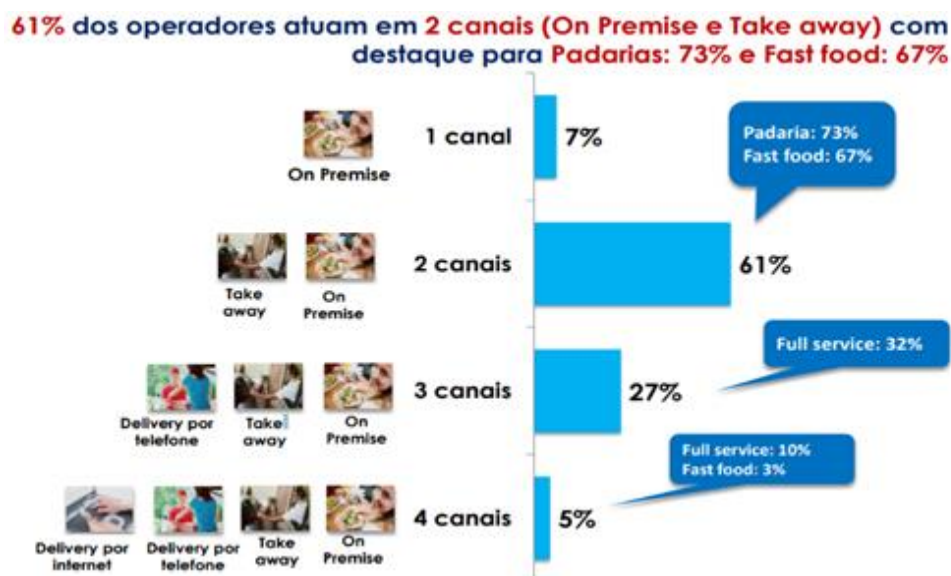
Uma série de fatores que influenciam o aumento desse serviço, o aumento populacional que segundo a Organização das Nações Unidas (ONU), a população mundial cresceu 170% de 1950 até 2009, e estima-se chegar em 8,1 bilhões de pessoas no planeta até 2025 a urbanização das cidades as migrações de pessoas das áreas rurais para áreas mais urbanizadas, torna a busca desse serviço maior, pois a pessoas deixam de produzir seu próprio alimento em propriedade rurais, buscando se alimentar com alimento mais rápido preparo. Outro fator é a mulher no mercado de trabalho e renda elevadas junto com a falta de tempo para produzir seu alimento, assim como o desconhecimento de preparo para fazer a refeição afeta nas escolhas desses usuários na busca do *FoodService*. (MACHADO, 2017)

2.2 PRINCIPAIS CANAIS DE ATENDIMENTO OFERECIDOS EM EMPRESAS *FOODSERVICE*

Segundo uma pesquisa feita pelo Instituto *FoodService* Brasil (IFB) 2018, (Figura 2), diz que 61% das empresas entrevistadas trabalham com dois canais de atendimento *on premise* e *take away*, principais empresas, padarias e *fast food*, 7 % das empresas que participaram da pesquisa trabalha com um canal de atendimento o serviço *on premise*.

As empresas que tem o serviço *full service* com mais de 3 canais de atendimento que tem delivery por telefone, *take away* e *on premise* representa 27 %, as empresas que tem 4 canais de atendimento são apenas 5% dos entrevistados na pesquisa trabalhando com todos os serviços como demonstrado na figura 2 (IFB, 2018).

Figura 2 - Pesquisa de canais de atendimento segundo IFB



Fonte: (IFB, 2018)

O estudo da IFB foi feito na data 10/02/2014 a 24/02/2014 e contou com 5000 mil consumidores e 1580 entrevistas pessoais com operadores de estabelecimento de *FoodService*, em todas as regiões do Brasil, norte, nordeste, centro-oeste, sudeste e sul do país (IFB, 2018).

2.2.1 Serviço de atendimento *On Premise*

On premise significa comer no local que o estabelecimento tem para fazer sua refeição. A grande maioria das empresas fornece esse tipo de atendimento onde o cliente tem que ir até o local da empresa e fazer seu pedido e depois fazer sua refeição. Isto corresponde ao sistema mais tradicional de atendimento, e nesse tipo de serviço sempre têm-se o deslocamento para chegar até o local e o tempo de espera para fazer o pedido. A grande maioria dessas empresas também apresenta o serviço de atendimento *take away* (IFB, 2018).

2.2.2 Serviço de atendimento *Take away*

Take away significa em fazer o pedido e levar seu pedido, ou seja, levar seu alimento para comer em outro local. Este tipo de serviço é bastante comum em estabelecimentos que estão em ruas onde não tem um local físico apropriado para se sentar e se alimentar ou em padarias. Esse tipo de serviço de atendimento tem as mesmas características de atendimento do *on premise* com a única diferença que o cliente não se alimenta no local (IFB, 2018).

2.2.3 Serviço de atendimento via telefone e *delivery*

Atendimento via telefone é uma forma de atendimento mais comum, o cliente com auxílio de um telefone fixo ou até mesmo com um celular liga para o estabelecimento para fazer seus pedidos para facilitar e adiantar o processo de agendar ou pedir sua refeição.

O serviço de *delivery* é quando a empresa oferece para o cliente a entrega de seu pedido feito por telefone em sua casa sem precisar que o cliente saia de casa para comer, esse é um serviço muito utilizado.

Mesmo com as facilidades desse tipo de serviço, ainda há alguns fatores que causam dificuldades nesse atendimento. Por exemplo, pode ocorrer de atender uma ligação ruim e fazer o pedido errado. As péssimas condições da telefonia móvel são um reflexo disso, linhas sempre ocupadas ocasionadas pelo grande volume de pedidos solicitados simultaneamente, agravante a isso os ruídos nas linhas dificultam ainda mais o atendimento (IFB, 2018).

2.2.4 Pedidos via internet

Hoje esse tipo de pedidos é a forma mais qualificada para se fazer os pedidos. Apenas com auxílio da internet você consegue fazer seu pedido de uma forma segura e rápida e alguns dos aplicativos existentes também oferecem a opção de entrega de seus pedidos.

Segundo dados da Associação Brasileira de Bares e Restaurantes (ABRASEL), o ritmo de crescimento do número de pedidos via aplicativos de *delivery* é de R\$1 bilhão a cada ano, um crescimento superior a 12%. Em 2015, as cifras superaram R\$9 bilhões e a expectativa é fechar 2017 acima dos R\$10 bilhões. (TERRA, 2017).

Desta forma, observa-se que as empresas estão se adaptando cada vez mais ao mercado e buscam reduzir seus prejuízos e aumentar seu lucro tornando a hora de fazer sua reserva ou pedido mais rápida e simples.

Para o cliente a comodidade é ainda maior, não necessita mais ligar para fazer o pedido, esperando que alguém lhe atenda, e nem precisa se deslocar até a empresa para comprar a sua refeição, havendo custo maior do seu alimento, por causa do deslocamento para esse trajeto. Com a utilização da internet nas empresas isso se tornou simplesmente usar um acesso a um site ou um aplicativo para fazer seu pedido.

2.3 INTERNET

Conceituando a internet de forma geral, pode-se afirmar que corresponde a uma coleção de redes conectadas uma com as outras por meio dos protocolos TCP/IP², sendo a maior rede de computadores do mundo (CRONIN, 1996)

O surgimento da internet ocorreu nos anos de 1960, então denominada Arpanet usada exclusivamente pelos militares, teve como objetivo interligar departamento de defesa norte-americanos e órgãos militares e universidades americanas. O sucesso do uso da internet deu-se a forma de comunicação entre todos os participantes da rede que era utilizada denominada *Internet Standards*, baseada em protocolos padrões de comunicação *Internet Protocol* (IP) (FRANCO JR., 2003).

Baseando-se em uma arquitetura de cliente/servidor onde um cliente usuário busca as informações em um servidor onde estão os dados, para fazer essa busca o cliente tem que utilizar *web browsers* (*software* de navegação), fazem a busca de dados nos servidores. (CAPITANIO, 2008).

Muitos confundem a *World Wide Web* (*Web*) sendo a própria internet mais não é, a *Web* e um mecanismo de transporte, a internet utiliza a *web* para transportar as informações de cliente/servidor, é padrão mundialmente aceito para armazenar, formatar, recuperar e exibir informações, a *Web* aceita todos os padrões de informações digitais, como texto, gráficos, som e hipermídias. (CAPITANIO, 2008).

Segundo Rodrigo Soares a independência da plataforma é o mais eficiente ambiente de desenvolvimento de aplicações, as empresas criam interações e diferentes relacionamentos com seus clientes usando a internet, criando novos modelos de negócios, sistemas para monitorar hábitos dos clientes. (SOARES; HOPPEN, 1997).

2.3.1 Negócios na Internet

Com o uso da internet as empresas estão modificando seus canais de venda, para uma internet onde as mesmas possuem uma autonomia muito maior. Na internet o produto pode ficar disposto 24hs por dia para ser comercializado, finais de semana e feriados, todos os dias, todos os 365 dias do ano, sem datas comemorativas e feriados, onde o produto está à disposição para o mundo inteiro ver, não existem limites geográficos e de fronteiras, podem ser vendidos mais rapidamente com menos esforços, com a possibilidade de atender muitas solicitações de uma única vez. (CAPITANIO,2008).

Observa-se que com os avanços das tecnologias o cotidiano está se tornando mais fácil as compras e serviços online, isso possibilita que as pessoas realizem atividades necessárias com mais eficiência. Proporcionando um maior tempo livre para se realizar outras tarefas como o lazer familiar. A internet facilita inclusive a acessibilidade as atividades de lazer, pois agiliza a construção de rotas planas e de viagem, também é possível comprar passagens aéreas, fazer reservas de hotéis entre outras. (CAPITANIO,2008).

Hoje as empresas estão focadas cada vez mais em vender seus produtos na internet, usando o comércio eletrônico, segundo Leonardo Almeida (2017), através do relatório do *Webshoppers* nº35, comenta que o comércio eletrônico no ramo de *FoodService* teve um aumento de 8% no ano de 2016, com um rendimento no valor de 44,4 bilhões de dólares, também nesse mesmo ano teve um aumento de 22% dos consumidores conectados, que utilizaram dispositivos móvel (*Tablets e Smartphones*), para fazer sua transação.

2.3.2 Comércio eletrônico (*e-commerce*)

Segundo Teixeira (2015) o conceito de comércio eletrônico pode ser definido como uma modalidade de compra a distância, em que a compra de um bem ou serviço é feita por algum tipo de equipamento eletrônico muitas vezes utiliza a internet para fazer essa negociação, nos quais esses equipamentos transmitem e recebem informações da transação, semelhantes as negociações feitas por catálogos, ou telefonemas, por programas de televisão, e ainda tantas outras.

Inicialmente o *e-commerce* era utilizado para vender produtos mais simples como, livros e CDs, e atualmente esse mesmo tipo de comércio é utilizado para vender todos os tipos de produtos e serviços (E-COMMERCE NEWS, 2017).

Atraindo muito a atenção das empresas por ser um grande facilitador de negócios, podendo vender seus produtos 24 horas por dia, todos os dias da semana, as empresas veem

neste tipo de comércio uma ferramenta para aumentar seus lucros e clientes sem grandes investimentos.

Segundo Bottentuit Junior e Pedroso (2012), consideram que o comércio eletrônico com as técnicas da *web* é uma grande ferramenta, para atrair mais clientes e aumentar o número de negócios e serviços e no Brasil tem inúmeras empresas que exploram esse tipo de atividade, algumas delas comercializam seus produtos em todo território nacional e outras tem o segmento regional, mas todos eles operando do mesmo modo, com o mesmo objetivo de proporcionar para os clientes as melhores ofertas com condições diferentes para aquisição de seus produtos via a internet.

Focado no início apenas no comércio para seu consumidor final, hoje o comércio eletrônico tem algumas categorias, por vendo a facilidade de como vender a empresas também começaram a vender suas matérias primas utilizando a internet, assim criando outros tipos de categorias e clientes que utilizam esse serviço.

Portanto segundo Bottentuit Junior e Pedroso (2012), foram criados seis tipos de conceitos de comércio eletrônico:

- *Business to consumer* (B2C) – essa categoria apresenta o comércio de um uma empresa com seus consumidores finais.
- *Business to business* (B2B) – esse tipo de comércio e feito entre empresas onde uma empresa comercializa seu produto ou serviço para outra empresa cliente.
- *Business to employee* (B2E) – nesse serviço a empresa comercializa seus produtos ou serviços com seus colaboradores (colaboradores).
- *Business to administration* (B2A) – é a empresa que disponibiliza seus produtos ou serviços para a administração pública, tendo outra nome que é *business to governemment* (B2G).
- *Consumer to consumer* (C2C) – nesse tipo o comércio é entre pessoas físicas, consumidor compra de outro consumidor.
- *Consumer to administration* (C2A) – nessa categoria o consumidor vende seu produto ou serviço para administração pública ou governamental.

Nessas seis categorias o tipo que mais se destaca é a B2C, e os produtos mais comercializados são passagens aéreas, livros e CDs, ingressos, tendo um grande aumento de vendas em outros produtos, com o aumento de empresas adentrando ao mercado eletrônico (BOTTENTUIT JUNIOR; PEDROSO, 2012).

2.3.3 *Business to Consumer (B2C)*

Esse tipo de comércio é focado aos consumidores finais, pode ser dinâmico e fácil, tem um maior desenvolvimento por ter um volume grande de empresas que entram para o comércio eletrônico utilizando a internet, hoje existem várias lojas virtuais e aplicativos de todos os segmentos, comercializando todos os tipos de bem de consumos como tênis, livros, *software*, automóveis, casas, produtos alimentares, produtos financeiros, serviços e publicações digitais, etc (FERNANDES, 2012).

Em consonância com Pereira e Prado (2016), o modo tradicional das empresas fazem seu relacionamento com cliente se rompe abrindo outros canais de vendas, as tecnologias vem abrindo uma interação mais rápida e dinâmica com o cliente, em um ambiente global e a era digital revolucionou e influenciou a sociedade o modo de interagir diariamente, as pessoas estão cada vez mais conectadas na internet e com as empresas utilizando o comércio eletrônico, essas empresas podem estar em qualquer lugar e qualquer tempo, estreitando a relação de empresa/consumidor, pensado como ter um melhor atendimento ao cliente consumidor, as empresas tem que ter estratégias de vendas para conseguir o sucesso no B2C, tendo uma boa administração em seu canal de venda, ter visual claro, boas condições de compra com ações comerciais competitivas.

2.4 SOLUÇÕES B2C PARA *FOODSERVICES*

Com o aumento da população e as inovações tecnológicas estão mais atualizadas a cada dia, as empresas *FoodServices* têm que dar suporte as todas essas transformações, investindo em inovações para melhorar e otimizar seus processos, controlando fluxo de caixa, evitar perdas, conhecer melhor seu cliente, e evitar ter perdas desnecessárias, para conhecer mais os clientes as empresas podem criar programas para fidelizar clientes, para otimizar suas vendas, podem criar vários canais de vendas, a tecnologia podem ser um grande aliada para essas soluções onde criando aplicativos móveis podem registrar informações dos clientes, criar cardápios on-line, fazer a gestão de pedidos pelo aplicativo, utilizando a favor da empresa essas inovações para conhecer melhor o cliente (CONTENT, 2017).

2.5 APLICAÇÕES MÓVEIS DE EMPRESA *FOODSERVICE*

Abaixo são descritas algumas aplicações de empresas *FoodServices* as mais conceituadas por usuários clientes consumidores.

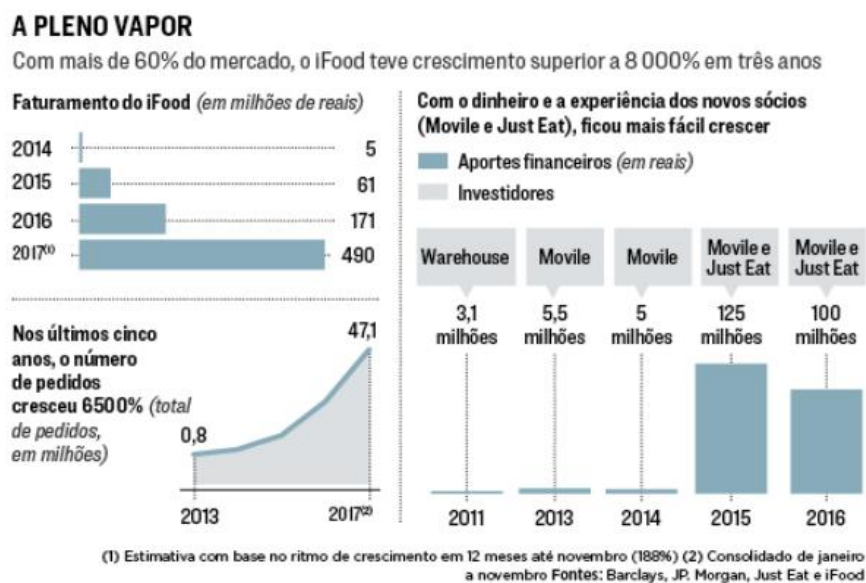
2.5.1 Ifood

Segundo Bertao (2018), o ifood (Figura 4) uma solução móvel para empresas do segmento de comidas e bebidas, é o aplicativo mais utilizado no país comparado aos seus concorrentes. Com o aplicativo é possível realizar o pedido de sua refeição e a entrega por *delivery*. O ifood funciona como um *markertplace*, que significa que são muitos restaurantes de vários tipos de comidas ou serviço que vende em um único local.

Onde você navega entre as empresas que fornecem o serviço de comida fora do lar, já cadastrada no sistema do ifood, e logo em seguida entra na página das empresas e escolhe o que quer comer e faz o pedido e escolhe a opção de ir buscar a refeição o entregar na sua residência na forma de *delivery*.

Seu crescimento no ano de 2017 comparado ao anterior foi de 188% e espera-se que no fim desse ano chega aos 490 milhões de reais em receita em pedidos feitos pelo APP, o Ifood em tem uma avaliação de 410 milhões de reais segundo o banco britânico Barclays (BERTÃO, 2018).

Figura 3 - informações da estimativa de faturamento do ifood no decorrer dos até o ano de 2017



Fonte: (BERTÃO, 2018)

Uma estimativa de crescimento astronômico como está na figura 3, o ifood em números de empresas cadastradas no seu início era de 230, hoje já passa das 22 mil empresas, o aplicativo faz uma quantidade de 5 milhões de pedidos por mês, atualmente no ifood são faturados 4 bilhões de reais em pedidos feitos nos aplicativos para as empresas vinculadas no aplicativo, atendendo em várias plataformas como mostrado na figura 4 (BERTÃO, 2018).

Figura 4 - Imagem do ifood em vários canais de vendas na internet



Fonte: (IFOOD, 2018)

2.5.2 Burger King

Teve origem nos Estados Unidos a rede de restaurantes em 1953, o Burger King sua primeira unidade era em Jacksonville na Flórida, no início tiveram dificuldades mais os fundadores sempre tiveram a ideia de levar comida para as pessoas com um baixo custo, hoje é a empresa e uma rede de restaurantes espalhada pelo mundo, sempre muito agressivo em suas campanhas de marketing com seu rival o Mc donald's (BURGER, 2006).

No Brasil o Burger King teve sua primeira franquia em 2004, com a mesma postura agressiva dos Estados Unidos colocando frases provocantes em suas propagandas, a primeira cidade no Brasil que a empresa entrou foi em São Paulo hoje tem mais de 100 pontos de vendas espalhadas em todo o país.

Atualmente a Burger King também lançou um aplicativo móvel (Figura 5), com a ideia de dar descontos em cupons promocionais para os usuários do App, o seu segundo objetivo do novo canal de venda e mostrar todo o menu que a empresa trabalha e também mostrar para o usuário onde fica o um Burger king mais próximos, para que possa levar seu

cupom promocional onde chegando no local ele mostra seu cupom pela tela do aplicativo e retirar seu pedido com descontos, gerando uma fidelidade com o cliente.

Figura 5 - Telas do Aplicativo do Burger King



Fonte: (APPTTEAM, 2018)

2.5.3 Alpha labs

A empresa Alpha Labs é uma empresa localizada em Santa Catarina na cidade de Porto Belo, inicialmente em 2011 quando fundada a empresa tinha o foco em desenvolver *softwares* para gestão até o ano de 2017 quando lançou um aplicativo de *delivery* agregando seu sistema de gestão com esse novo App. Na figura 6 pode se observar as telas desse aplicativo que é voltado para empresas *foodservices*, a Alpha Labs aluga o serviço de gestão de pedidos junto com aplicativo ou uma solução de um site *web*, modelando o aplicativo para cada perfil de empresa *FoodServices* (ALFALABS, 2018).

Seus números por ser uma empresa pequena e está em ascensão, sendo que no seu primeiro ano a solução de *delivery* já deram mais de 5 milhões em transações com mais de 80 mil pedidos feitos para empresas contratantes dessa solução, tendo mais de 30 mil usuários cadastrados no aplicativo para fazer pedido.

Figura 6 - Telas do aplicativo desenvolvido pela ALFALABS para empresa *Foodservices*



Fonte: (ALFALABS, 2018)

3 ESTUDOS DAS TECNOLOGIAS PARA O DESENVOLVIMENTO DA SOLUÇÃO EM ESTUDO

Nesse capítulo serão abordados alguns tipos de linguagens e ferramentas utilizadas para criar a soluções para empresas do tipo *FoodService*.

3.1 SISTEMA ANDROID

O Android surgiu em 2003 em Palo Alto na Califórnia, originalmente com a ideia de seus criadores era se tornar um sistema inteligente para câmeras fotográficas digitais, mas percebendo que o mercado desse tipo de produto não tinha uma amplitude boa de mercado, seus criadores Andy Rubin, Rich Miner, Nick Sears e Chris White, mudaram o foco e passaram a desenvolver para dispositivos móveis (MEYER, 2017).

Sendo secreto seu desenvolvimento no início, quando mudaram o foco para aparelhos móveis, Rubin e sua equipe ofereceram um sistema inovador, com um novo jeito de utilização para aparelhos móveis, virou uma solução *open source* baseada no *kernel Linux*, com uma interface simples, funcional e entregava vários outros instrumentos (MEYER, 2017).

Em 2005 segundo Meyer (2017) o sistema Android foi comprado pela empresa Google assim nasceu a *Google Mobile Division*, era uma divisão de pesquisa móvel da maior empresa de tecnologia do mundo, uma dúvida inicialmente o sistema Android, pois tinha concorrência e era comparado com os outros sistemas como o *Windows Mobile* e *Ios* da Apple, mais em 2006 teve o primeiro aparelho móvel com Android, e no ano seguinte as grandes montadoras de aparelhos móveis começaram a usar o sistema Android para rodar nos seus aparelhos.

Hoje é o sistema operacional mais utilizado no mundo, em 2014 cerca de 81,5 % de aparelhos utilizados no mundo continha o sistema Android, mais de 3 milhões de aparelhos com o sistema foram comercializados no mundo, devido ser uma plataforma *open source* tem mais de 11.868 tipos de Android, rodando em 8 versões de Android diferentes (MEYER, 2017).

Tipos de versões de Android e seu ano de lançamento são:

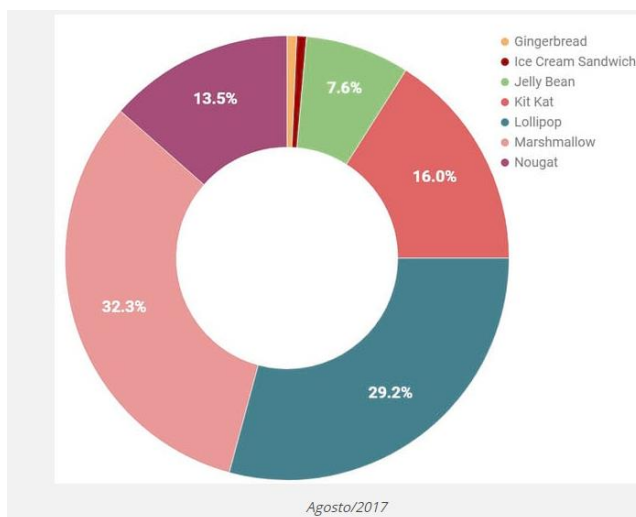
- Android Alpha e Beta (2007 ~ 2008)
- **Android 1.0 – Astro (2008)**
- Android 1.1 – Battenberg (2009)
- Android 1.5 – Cupcake (2009)
- Android 1.6 – Donut (2009)

- **Android 2.0 (2.0.1 e 2.1) – Éclair (2009 ~ 2010)**
- Android 2.2 (2.2.1; 2.2.2 e 2.2.3) – Froyo (2010)
- Android 2.3 (2.3.1; 2.3.2) – Gingerbread (2010 ~ 2011)
- Android 2.3.3 (2.3.4; 2.3.5; 2.3.6 e 2.3.7) – Gingerbread (2011)
- **Android 3.0 – Honeycomb (2011)**
- Android 3.1 – Honeycomb (2011)
- Android 3.2 (3.2.1; 3.2.2; 3.2.3; 3.2.4; 3.2.5 e 3.2.6) – Honeycomb (2011 ~ 2012)
- **Android 4.0 (4.0.1; 4.0.2; 4.0.3 e 4.0.4) – Ice Cream Sandwich (2011 ~ 2012)**
- Android 4.1 (4.1.1 e 4.1.2) – Jelly Bean (2012)
- Android 4.2 (4.2.1 e 4.2.2) – Jelly Bean (2012)
- Android 4.3 (4.3.1) – Jelly Bean (2013)
- Android 4.4 (4.4.1; 4.4.2; 4.4.3 e 4.4.4) – KitKat (2013 ~ 2014)
- Android 4.4W (4.4W.1 e 4.4W.2) – KitKat (2014)
- **Android 5.0 (5.0.1 e 5.0.2) – Lollipop (2014)**
- Android 5.1 (5.1.1) – Lollipop (2015)
- Android 6.0 (6.0.1) – Marshmallow (2015)
- Android 7.0 (7.1, 7.1.1 e 7.1.2) - Nougat (2016)
- Android 8.0 (8.1) - Oreo (2017)
- Android 9.0 – Pie (2018)

Essas são todas as versões de Android até outubro de 2018 ANDROID INC, 2018).

Na Figura 7, mostramos o gráfico de utilização em agosto de 2017 dos tipos de versões ainda em uso (MEYER, 2017).

Figura 7 - Gráfico de utilização de versões do Android



Fonte: (MEYER, 2017)

Com toda evolução o sistema Android, hoje já está em outros tipos de tecnologias como: automóveis, relógios, videogames, entre outros se tornando o sistema que está presente em vários meios tecnológicos tornando possível modelar para cada tipo de *hardware*.

A Google utiliza a Google Play que é a biblioteca de aplicativos mais conhecida para o sistema Android, servindo para auxiliar a distribuição de aplicativos, as empresas que desenvolvem soluções de sistemas para Android podem fazer o *upload* e gerenciar seus aplicativos, a Google Play tem como objetivo ter um local comum para os desenvolvedores colocar seus aplicativos, para ter um aplicativo na loja é necessário pagar uma taxa de entrada de 25 dólares e a Google retém 25 % da venda dos aplicativos, dando 75% da venda para os desenvolvedores (LECHETA, 2013).

3.2 INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

São conjuntos de ferramentas para o desenvolvedor com objetivo de facilitar na hora de projetar seu aplicativo ou projeto, serve para programar, projetar, depurar, um aplicativo ou qualquer tipo de *software*, a IDE deve fornecer uma sintaxe colorida que ao escrever pode auto completar códigos, facilitando a escrita e também mostrar erros de sintaxe e marcação, uma IDE tem que suportar linguagens de marcação como, por exemplo, no Android é o XML, e a linguagem de programação que pode ser o JAVA ou o Kotlin (FREDERICK; LAL, 2011).

3.2.1 Desenvolvendo aplicativos para Google usando o sistema Android

Para desenvolver aplicativos para o sistema Android podem ser utilizadas diferentes linguagens de programação, a Google tem quatro linguagens de programação que podem ser executadas na sua IDE, Java, kotlin, C/C++ que são caracterizadas como linguagens nativas para a programação Android, suportando várias integrações com API's (*Application Programming Interface*) internas da própria Google e externas para desenvolver os aplicativos.

A API para a programação significa ter interação com outros sistemas já desenvolvidos por empresas, assim como padrões e rotinas de aplicações, códigos já existentes desenvolvidos que dão acesso a uma plataforma *web* ou um aplicativo.

A Google tem uma comunidade de desenvolvedores enorme, que fornecem API's para diversos tipos de serviços e solução, oferecendo como API's de mapas, áudios e vídeos. Integrando com diferentes linguagens de programação, hoje no mercado, a API mais conhecida

da Google é o Google Maps, utilizada em diversas aplicações, pois oferece o serviço de localização no mapa, são utilizadas em sites, celulares, relógios, carros e mais.

Os aplicativos para Android podem ser desenvolvidos com outras linguagens de programação, sendo outras empresas que darão o suporte no desenvolvimento, não será a Google, a Microsoft em 2015 lançou a IDE Visual Studio para desenvolver aplicativos para Android, usando a linguagem de programação C++, a Microsoft quer trazer uma adaptação dos aplicativos do *Windows Mobile* para o sistema Android (LUCENA, 2018).

Para desenvolvermos para o sistema Android é preciso ter um emulador de ferramentas do sistema, a emuladora irá simular um aparelho Android, sendo necessário ter instalado do computador junto com a IDE o emulador Android SDK (LUCENA, 2018).

3.2.2 Android SDK (*Software development kit*)

Android SDK é um *software* utilizado para emular um sistema Android, onde simula um celular com o sistema, possibilitando a conversão para diversas opções de versões do Android, junto com conjuntos de ferramentas e API's da linguagem JAVA ou Kotlin, com esse emulador é possível iniciar um celular dentro da própria IDE ou conectar um celular pelo cabo USB e testar o seu desenvolvimento no próprio celular (LECHETA, 2013).

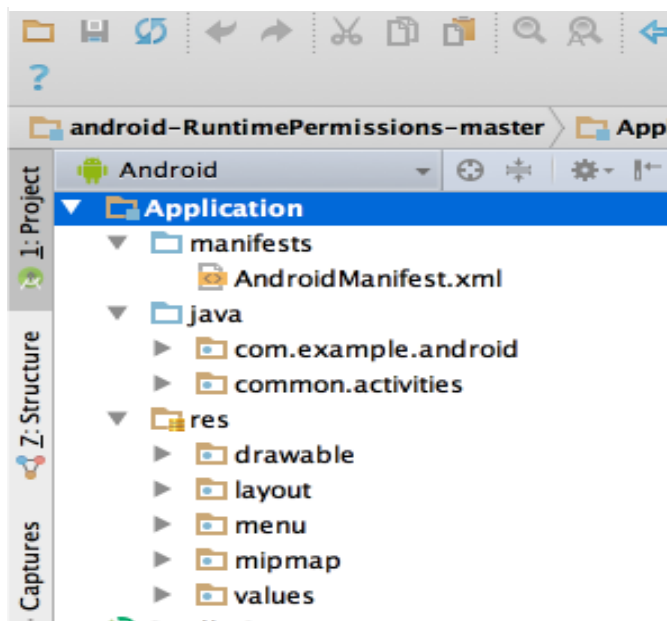
3.2.3 IDE Android Studio

Segundo Android Inc (2018), hoje a IDE oficial para desenvolvimento de aplicativos e Android Studio, baseada em outra IDE, na IntelliJ IDE, oferecendo todas as ferramentas e códigos da IntelliJ, no Android Studio se tornando uma solução mais completa, contendo mais recurso e aumentando produtividade dos desenvolvedores. A estruturação da IDE é feita em módulos de arquivos-fontes e recursos, incluindo módulos dos aplicativos Android, módulos de biblioteca, módulos do Google *APP Engine*. A IDE apresenta os seguintes módulos (Figura8):

Manifestos: contém o arquivo *AndroidManifest.xml*

- Java: contém arquivos de código fonte do Java.
- Recursos: Contém todos os recursos que não são códigos, como layouts XML, *Strings* de IU e imagem *Bitmap*.

Figura 8 - Android Studio Estrutura de arquivos



Fonte: (ANDROID INC, 2018)

3.2.4 *Manifest* -Android Manifest

Segundo Android Inc (2018) o arquivo *Manifest* é obrigatório para quem está desenvolvendo para Android, contendo no arquivo androidManifest.xml, toda a informação essencial sobre o aplicativo serve para:

- Dar nome aos pacotes Java no aplicativo, esses nomes são os identificadores exclusivos do APP.
- Descrever cada componente do APP, sua atividade, serviços, receptores de transmissão e conteúdo XML que contém no aplicativo.
- Estabelecer os processos de hospedar cada componente.
- Revelar cada permissão que o APP pode acessar em partes protegidas da API e Integração com outros aplicativos.
- Expor o a API mínima do Android.
- Mostrar as bibliotecas as quais são veiculadas o aplicativo.
- Listar as bibliotecas às quais o aplicativo deve se vincular.

3.2.5 Módulo de classes JAVA

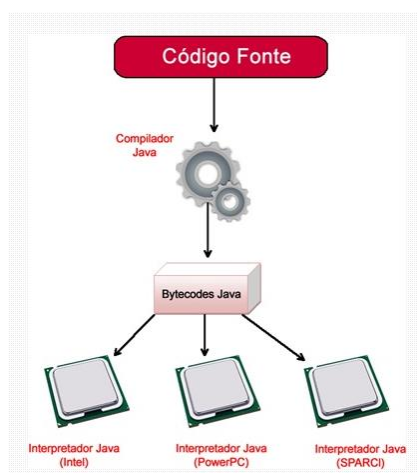
No Android Studio o *java package*, utilizado para separar todas as classes Java que são feitas no desenvolvimento dos aplicativos, toda parte de *back-end* e interações com outros aplicativos, comunicação para bancos de dados e comandos de ações como mudar de layouts são feitas na classe Java.

A linguagem JAVA foi fundada no ano de 1991 por uma equipe de engenheiros liderados por Patrick Naughton, Sun Fellow e James Gosling, a ideia principal da linguagem para que aparelhos eletrônicos se comunicassem entre si, mais para isso antes teriam que inventar uma linguagem que não dependesse do processador, pois os clientes não estavam interessados no que utilizavam em *hardware* e sim na tecnologia. (FILGUEIRAS, 2015).

Assim surgiu o sistema GreenOs, onde criaram uma linguagem de programação batizada de OaK (carvalho), logo depois mudaram o nome da linguagem para JAVA homenageando a ilha da Indonésia de onde importavam café, que era consumido pela equipe. (FILGUEIRAS, 2015).

Vinícius (2018) diz que o Java possui nas suas características o poder de desenvolver o código em Java sem se preocupar com o *hardware*, pois ele é independente do sistema operacional para ser executado, isso acontece por que quando é executado um código Java ele é emulado por uma máquina virtual JVM (*Java Virtual Machine*) e depois de emulado é gerado um *Bytecodes* Java e isso é enviado para a linguagem de máquina para a execução dos sistemas. Na figura 9 pode se observar com é seu funcionamento.

Figura 9 - Imagens de como é compilado um código em Java.



Fonte: (VINÍCIUS, 2018)

3.2.6 Recursos Layout XML Android

O recursos de layout fornecem a estrutura visual do aplicativo, e a interface do usuário IU (*User Interfaces*) é possível ser declarado de duas formas:

- Mostrar em Elementos XML: o Android um dicionário XML onde tem todos os grupos de *View* subclasse de *View*, layouts e *Widgets*.
- Instanciar de elementos do layout em tempo de execução: é uma criação de objetos de *View* e *Viewgroup* e processa essas propriedades.

Podendo usar a interação entre essas duas maneiras de fazer o layout em XML, as pastas de recursos também são colocadas as imagens, padrões de *Strings* e modelos de estilos para facilitar na hora do desenvolvimento (ANDROID INC, 2018).

Ainda segundo Android Inc (2018), as vantagens para utilizar o XML separados e para melhor apresentar os códigos no desenvolvimento e sendo possível modificá-los e adaptá-los sem mexer no seu código fonte.

3.3 WEB SERVICES

Silva (2014) diz que *web services* são aplicativos que são hospedados na internet com o intuito de enviar e receber dados, é necessário enviar resposta para quem fez a requisição, um sistema de comunicação de duas estruturas de dados diferentes, que são traduzidas em uma linguagem universal como XML ou Json.

Allan (2014) diz que *web services* é definido por uma comunicação entre *software* independente da sua plataforma de desenvolvimento, o que torna marcante esse serviço que a comunicação sempre feita na rede (internet) e sempre ter que estar disponível na *web*.

Allan (2014) também menciona que a importância desse tipo de serviço é permitir a integração das plataformas corporativas, e desenvolver soluções descompactadas podendo desenvolver soluções externas para interação de servidor e dispositivos móveis.

A forma de comunicação entre essa plataforma onde esses dados trafegam são feitas por protocolos, sendo o mais utilizado o protocolo IP (*Internet Protocol*), é possível que os dados trafeguem corretamente sem perdas de dados (SILVA, 2014).

Silva (2014) menciona que o IP é composto de duas partes, sendo a primeira onde identifica a rede que o equipamento está conectado, e a segunda serve para identificar a própria rede, trabalhando em conjunto com outro protocolo o TCP (*Transmission Control Protocol*) formando o TCP/IP padrão de comunicação utilizado na Internet.

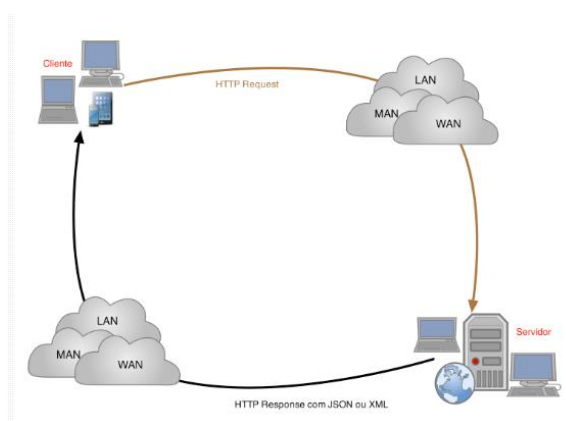
Silva (2014) também aponta a necessidade de outro protocolo para que um site seja exibido pela rede o protocolo HTTP (*Hyper Text Transfer Protocol*) o qual é destinado fazer a transferência de arquivo na rede sendo ligado no serviço cliente/servidor por uma URL (*Uniform Resource Locator*).

3.3.1 Formas de transmissão de dados *Web Services*

Para a transmitir dados entre as plataformas têm duas maneiras que são o *Representational State Transfer* (REST) e o *Simple Object Data Protocol* (SOAP) (SILVA, 2014).

- REST, tem o funcionamento da seguinte forma (Figura 10) o cliente envia uma requisição HTTP para o servidor com informações necessárias para uma execução de algum comando. O servidor processa a requisição e retorna para a cliente uma resposta HTTP contendo um arquivo XML havendo nessa mensagem uma simples informação ou um conjunto de dados muitos complexos (ALLAN, 2014).

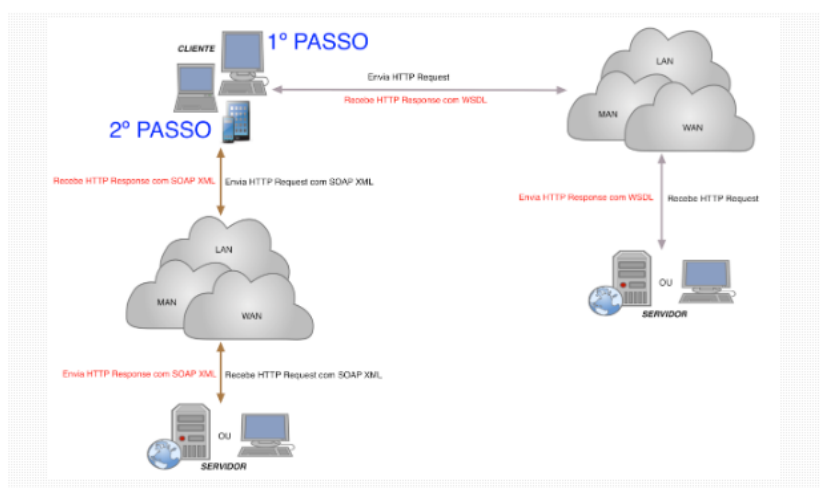
Figura 10 - Diagrama de funcionamento de um *Web Service* utilizando REST



Fonte: (VINÍCIUS, 2018)

- SOAP, o primeiro passo o cliente é localizado em um componente eletrônico como um computador, notebook ou um dispositivo móvel e é enviado uma requisição (HTTP) para o servidor por meio da rede e o mesmo retorna um arquivo WSDL (*Web Services Description Language*). Onde esse arquivo no formato de XML, informa ao cliente como funcionará o serviço de comunicação sendo nomes de métodos, nome de parâmetros, e endereços do serviços padrão tanto para entrada ou saída de informação, a partir disso o cliente sempre fará as requisições (HTTP) em XML ou Json para o servidor processar e retornar outro XML ou Json com informações de dados simples ou muito complexos (ALLAN, 2014). Na figura 11 pode se observar melhor com funciona uma requisição por meio do SOAP.

Figura 11- Diagrama de funcionamento de uma *web Service* utilizando SOAP



Fonte: (VINÍCIUS, 2018)

3.4 SERVIDORES DE BANCOS DE DADOS

Banco de dados correspondem a um repositório de dados que são relacionados entre si, os dados podem ser de qualquer tipo como nomes, números, telefones, imagens, áudios, vídeos (RAMEZ ELMASRI SHAMKANT B. NAVATHE, 2005).

Ramez Elmasri Shamkant B. Navathe (2005) afirma que o banco de dados tem um papel essencial nas soluções utilizadas para computadores ou *software* que utiliza sua solução na internet, são utilizados em todas as áreas, nas engenharias, no comércio eletrônico, nas áreas de saúde e nas humanas.

No cotidiano, utilizamos diversas vezes o banco de dados nas nossas interações com componentes eletrônicos como pesquisar na internet, um saque de dinheiro no banco, uma reserva de um hotel ou restaurantes, utilizando a *web* ou componentes físicos como caixas eletrônicos, sempre encontraremos bancos de dados para armazenar e gerenciar os dados de cada site ou sistemas (RAMEZ ELMASRI SHAMKANT B. NAVATHE, 2005).

Um banco de dados pode ser controlado manualmente ou pode ser automatizado por computadores com gerenciadores de bancos de dados, os SGBD (Sistema Gerenciador de Banco de Dados), tem como objetivo a facilitação e de criar manipular e compartilhar os dados que são armazenados nos bancos de dados (RAMEZ ELMASRI SHAMKANT B. NAVATHE, 2005).

3.5 MYSQL (SGBD)

MySQL é um sistema que gerencia bancos de dados, se originou em 1996 com Michael Widenius e David Axmark, é um sistema de gerenciamento que tem o código aberto, não precisa adquirir o sistema para utilizá-lo na aplicação, muito popular por ser uma solução gratuita para gerenciamento mais também possui outra característica que faz ser popular, e disponibilizados em diferentes plataformas (SILVA; TOFFOLO, 2016).

Silva e Toffolo (2016) cita principais recursos do MySQL, tais como:

- Ambiente multiplataforma, tem apresenta interação com vários tipos de equipamentos com sistemas diferentes;
- Tem um alto desempenho e estabilidade;
- Sem limitação de acesso de usuário, pode ter conexões simultâneas;
- Leve, não precisa de um hardware poderoso para ser executado;
- Flexibilidade de aumentar de tamanho de capacidade se necessário.

Com a utilização do banco de dados e os avanços computacionais o desenvolvimento de solução precisa de uma linguagem de programação com o objetivo de especificar os controladores e as atividades do banco de dados.

3.6 A LINGUAGEM PHP

Foi criada em 1994, por Rasmus Lerdorf e tinha a necessidade de acompanhar seu currículo que foi hospedado na internet, em seguida teve várias atualizações nas suas versões e no ano de 1998 a linguagem passou a ter interações com diversos bancos de dados. Hoje o PHP está na versão 7 e também se tornou uma linguagem com orientação a objetos depois dos anos 2000 (SILVA; TOFFOLO, 2016).

O PHP é uma linguagem de programação para ambientes *web* que permite criar funções, variáveis, fazer *loops* em seu desenvolvimento sendo utilizado por milhões de sites. A linguagem PHP tem seu código aberto e toda sua documentação está disponibilizada no seu próprio site oficial, outra característica é de poder ser embutida na linguagem HTML, e apresentar a portabilidade para todos os sistemas operacionais, com atualizações frequentes da linguagem se adaptando ao mercado e as novas tecnologias (NIEDERAUDER, 2004).

3.7 A LINGUAGEM DE MARCAÇÃO HTML (*HIPERTEXT MARKUP LANGUAGE*)

Com o surgimento da *Web*, houve a necessidade de criar um linguagem que fosse entendida por diferentes meios. Tim Berners-Lee criou o HTML, com objetivo de sanar essa necessidade da *web*, e 1990 Marc Andreessen desenvolveu o primeiro *browser* onde popularizou o HTML (EDUARDO, 2012).

Baseado em conceito de hipertexto, que são conjuntos de elementos ligados por conexões, como palavras, imagens, vídeos, áudios, documentos etc., quando ligados entre se e formam uma grande rede de internet (EDUARDO, 2012).

HTML é uma linguagem de marcação na *Web*, um conjunto de códigos que pode ser desenvolvido em qualquer IDE, e são esses códigos que fazem a exibição do conteúdo de um site, podendo conter referências para imagens, vídeos, texto e áudios. O seu código só pode ser exibido por um navegador *web browser*, o qual tem a responsabilidade de analisar o código HTML e exibir o conteúdo em uma página *web* (SILVA; TOFFOLO, 2016).

3.8 JAVA SCRIPT

Uma linguagem que apresenta uma funcionalidade mais complexa nas páginas *web*, permite implementar animações, mapas mais interativos, gráficos 2D/ 3D e mostrar dados em tempo real. Isso só possível pela a utilização do Java Script nas páginas *web*, tornando elas menos estáticas e com mais interatividade (MOZILLA E COLABORADORES INDIVIDUAIS, 2018).

Seu desenvolvimento tem vínculo com a HTML, pode ser desenvolvido seu código em qualquer IDE que suporte à linguagem, o site construído com o Java Script tem uma interação maior com o usuário.

3.9 *CASCADING STYLE SHEETS*

O nascimento do CSS se deu para separar a linguagem de marcação o HTML e o estilo da página como cores, estilos de objetos, onde se dá agora usando o CSS para dar a aparência para uma página *web*, proporcionando mais flexibilidade e controle nas codificações estrutural de uma página (TABLELESS, 2018).

4 PROJETO DO APLICATIVO DE *FOODSERVICE*

Nesse capítulo será apresentado a empresa base para este trabalho de conclusão de curso, a análise da situação atual na empresa para agendamento de pedidos de comida, assim como a proposta da interface de navegação do aplicativo.

4.1 CANTINA DA TIA GRAÇA E TIA GRAÇA SALGADINHO

A empresa teve início em 2003 quando a dona Maria das Graças de Borba mais conhecida por Tia Graça começou a trabalhar com a produção de refeições para vender na lancheria da Escola Jovem, na cidade de Sombrio em Santa Catarina, localizada na região AMESC.

No início da empresa a Tia Graça trabalhava com produção de salgados fritos, bolos, cachorro quente e mini-pizza vendendo esses produtos no intervalo de cada período da escola. Na Escola Jovem a Tia Graça era responsável pela venda de mais de 120 salgadinhos por período, Matutino, Vespertino e Noturno, juntos esses períodos davam mais de 350 salgados grandes vendidos por dia.

Depois de alguns anos trabalhando na Escola Jovem os clientes que consumiam os salgados grandes começaram a fazer pedidos para produzir mini-salgadinhos para festas de aniversários e confraternização de seus filhos ou conhecidos, assim surgiu a ideia de atender à solicitação dos clientes e a proprietária iniciou a produção desse tipo de alimento. Com o passar dos anos foram se tornando cada vez mais frequentes os pedidos de mini-salgadinhos, se tornando um dos carros chefes de vendas da empresa, seguido pelas vendas de salgados grandes nos intervalos de cada período.

No ano de 2009 a empresária parou de vender salgadinhos na escola e montou um local próprio para vender seus produtos. Localizada na Rua Teodoro Rodrigues de Oliveira N° 463 Neste estabelecimento começou a produzir marmitas ao meio dia e salgadinhos para festas. A partir da estruturação de um local físico surgiram novos desafios e novos problemas, já que não haviam mais clientes fixo, como antes nas escolas onde os alunos se deslocavam para a cantina para fazer a refeição diariamente.

No ano de 2009 a empresa se tornou uma empresa que entregava o almoço em casa usando o *delivery* para fazer as entregas, podendo ser denominada *FoodService*.

A Cantina da Tia Graça desde então aumentou seus números de clientes e pedidos tendo que se dividir em duas empresas a Cantina da Tia Graça Salgadinhos e Cantina Tia Graça Restaurante. Na Cantina Tia Graça Restaurante montou-se uma nova base para atender apenas pedidos de marmita e aos clientes que desejam fazer sua refeição no local. Na empresa Cantina Tia Graça Salgadinhos produz-se apenas alimentos como salgadinhos para festa e confraternização, atendendo mais de 100 pedidos por mês.

Figura 12 - Cantina da Tia Graça 2009



Fonte: Google Street View

Figura 13 - Imagem da Cantina da Tia Graça Restaurante ano 2018



Fonte: Foto tirada pela equipe

Na Figura 12 é mostrado o primeiro local da Cantina Tia Graça onde era realizado a produção de alimentos, refeições prontas do tipo marmita e salgadinhos. O estabelecimento apresentava horários de funcionamento diferentes, ao meio dia vendia refeições trabalhando com o conceito de *take way e delivery*, e no período da tarde e finais de semanas entregava os pedidos de salgadinhos feitos pelo cliente só no modelo *take way*. Já na Figura 13 é mostrado o local de atendimento da Cantina Tia Graça em 2018.

4.2 SITUAÇÃO ATUAL DA CANTINA DA TIA GRAÇA RESTAURANTE E TIA GRAÇA SALGADINHOS

As duas empresas hoje em Sombrio são bem consolidadas em termos de números de clientes e de pedidos diários, com quase de 10 anos de funcionamento. Com o objetivo de se atualizar ao longo da sua história a empresa entrou em redes sociais para permitir interatividade com seus clientes como Facebook e Instagram, mas seu principal canal de venda hoje em dia ocorre por meio de ligações telefônicas, seguido por pedidos feitos no próprio local

Observando as duas empresas Tia Graça salgadinhos e a Cantina da Tia Graça restaurante, notou-se que a empresa que tem mais dificuldades na hora de fazer o pedido é a Tia Graça salgadinhos a qual apresenta várias dificuldades para fazer a gestão dos seus pedidos.

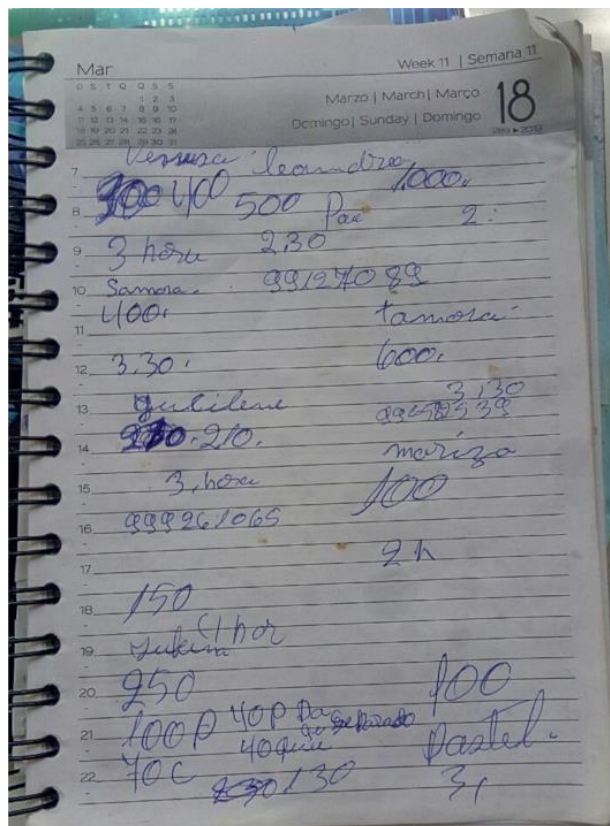
4.3 PROBLEMAS ENCONTRADOS NA GESTÃO DE PEDIDOS NA CANTINA TIA GRAÇA SALGADINHOS

A partir de uma análise presencial da forma como os pedidos são realizados atualmente foram identificados alguns possíveis problemas com o sistema de solicitação de pedidos por meio do telefone.

As seguintes dificuldades foram encontradas na solicitação de pedidos por meio do telefone: linha ocupada durante a tentativa de ligação, não conseguir atender o telefone na hora da chamada, dificuldades na linha como falha na comunicação, além de ser cansativo dado a necessidade de ter que explicar em todas ligações os tipos de salgadinhos que são produzidos para que o cliente escolha e faça a encomenda.

Depois do cliente definir os itens a serem pedidos o funcionário tem que armazenar as informações. Neste momento surge outro problema, pois a Tia Graça como é conhecida a dona da empresa, é alfabetizada, mas só estudou até a quarta série do Ensino Fundamental, e tem grandes dificuldades na hora de escrever os pedidos na sua agenda (Figura 14).

Figura 14 - Agenda de pedido Tia Graça Salgadinhos



Fonte: foto tirada pelos autores

Na Figura 14 pode-se observar que as informações referentes ao pedido não estão claras e não seguem um padrão para a fácil identificação de quais tipos de salgadinhos foram solicitados no pedido.

Em perguntas feitas para a Tia Graça na visita a empresa, questionamos ela sobre se já houve confusão na hora do pedido e quais foram? Ela respondeu que já houve confusão em vários momentos, tais como por não ter anotado o pedido certo, ou por ter realizado um erro na data de entrega do pedido.

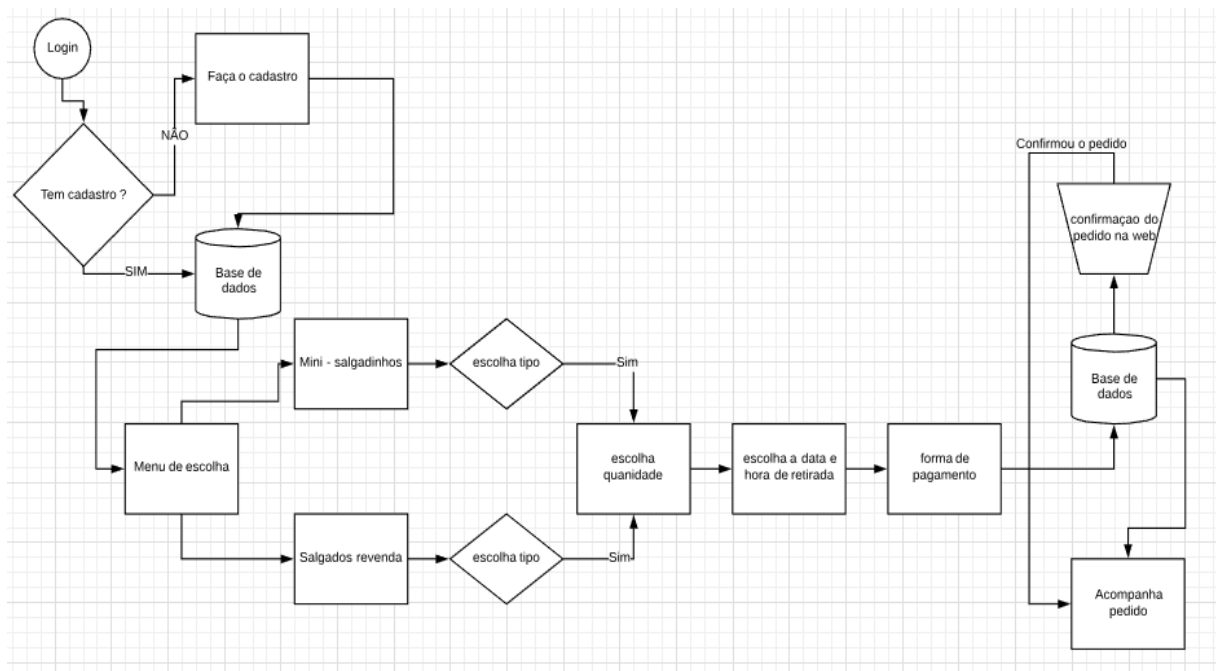
A Tia Graça comentou também que já houve vários problemas relacionados a clientes não aparecerem para pegar a encomenda e de não poder entrar em contato por não ter um registro do nome ou do telefone de quem realizou o pedido.

4.4 PROPOSTA DE UM SISTEMA DE AGENDAMENTO DE PEDIDOS NA TIA GRAÇA SALGADINHOS

A partir da análise realizada e da observação dos problemas com o sistema atual, decidiu-se criar um aplicativo para dispositivos móveis que contenha o sistema Android assim como uma solução que exija apenas a conectividade com a internet para fazer o pedido por meio de um *web browser*.

Na Figura 15 é mostrado um fluxograma de funcionamento da solução proposta.

Figura 15 - Fluxograma de funcionamento da solução proposta



Fonte: Desenvolvido pelos autores.

4.5 PROPOSTA DAS TELAS DE NAVEGAÇÃO DO APLICATIVO

Apresenta-se a seguir a proposta das telas de navegação do aplicativo.

4.5.1 Tela de login

Figura 16 - Tela de Login



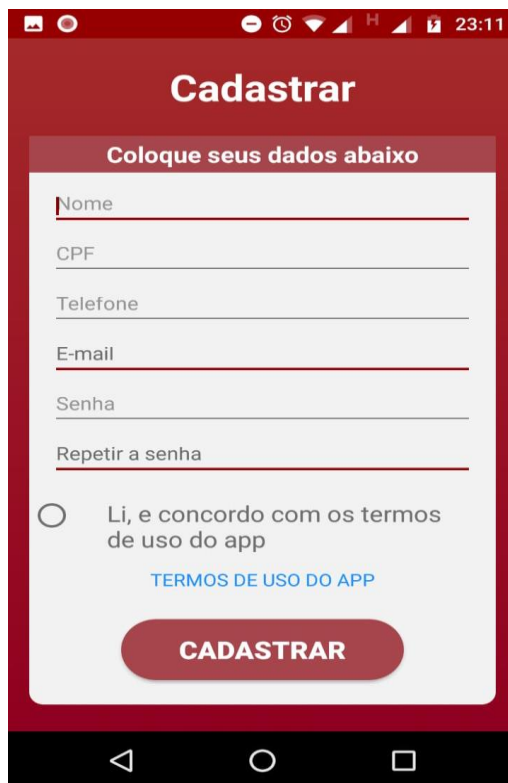
Fonte: Desenvolvida pelos autores.

Na primeira tela (Figura 16) pergunta-se ao usuário se ele já possui um cadastro no App. Se o usuário não possuir um cadastro o mesmo terá que efetuar um cadastro para acessar as informações do App.

4.5.2 Tela cadastro

Na tela de cadastro (Figura 17) utiliza-se como base os aplicativos existentes para solicitação de pedidos de *FoodService*.

Dados como do usuário, tais como Nome, CPF, Telefone, E-mail e Senha são armazenados no banco de dados. A partir da entrada de dados o usuário poderá acessar as próximas telas de navegação do aplicativo.

Figura 17 - Tela de Cadastramento

Cadastrar

Coloque seus dados abaixo

Nome

CPF

Telefone

E-mail

Senha

Repetir a senha

☐ Li, e concordo com os termos de uso do app

[TERMOS DE USO DO APP](#)

CADASTRAR

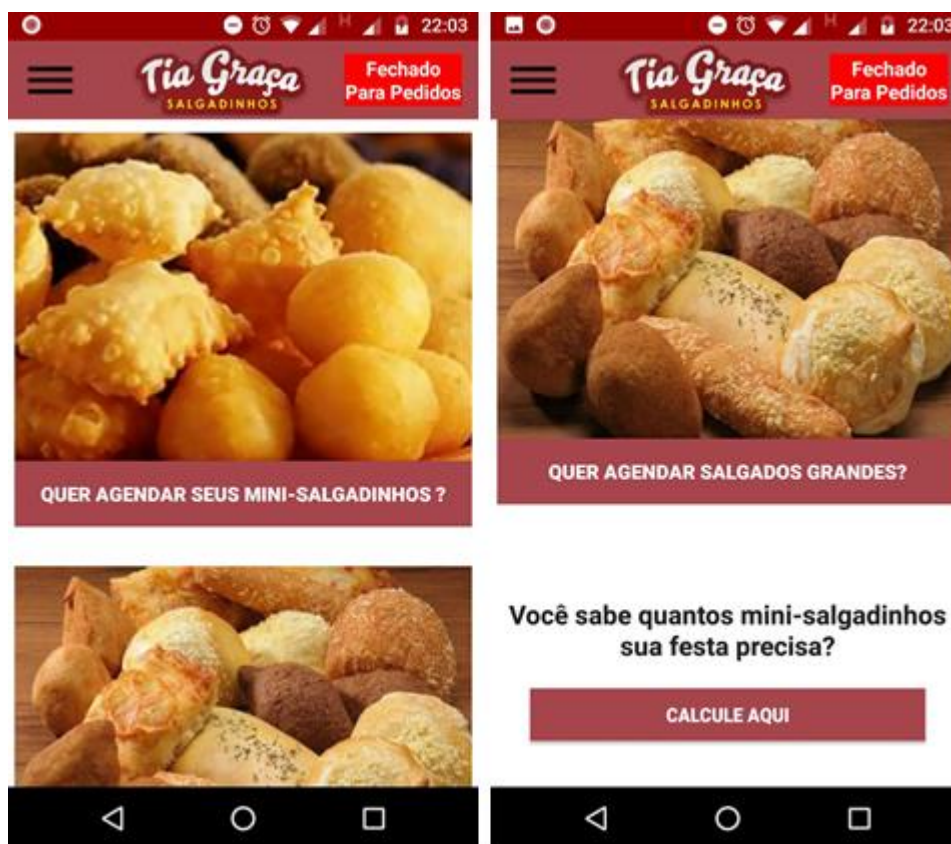
Fonte: Desenvolvida pelos autores.

4.5.3 Tela de menu

Na segunda tela (Figura 18), são apresentadas as opções de serviço que a Tia Graça Salgadinhos oferece para o usuário. Na primeira imagem à direita temos a primeira opção de selecionar pedidos para mini-salgadinhos, uma tela *Scroolview*, tela de rolagem. Na parte de baixo da tela mostra-se outra opção de serviço para o usuário, para agendar salgados grandes e por último no fim da tela colocamos um calculador de quantidades de mini-salgadinhos. Em relato a Tia Graça diz que sempre os usuários fazem essa pergunta “Quantos mini – salgadinhos em média vai na minha festa?” observando essa pergunta adiciona-se uma calculadora para calcular a quantidade de mini-salgadinhos para festa.

Na parte superior temos o *nav bar* onde indica-se para o usuário o horário de atendimento da Tia Graça salgadinhos e indica-se também o atual status da empresa (empresa aberta ou fechada).

Figura 18 - Tela Menu App Tia Graça Salgadinho



Fonte: Desenvolvida pelos autores (2018).

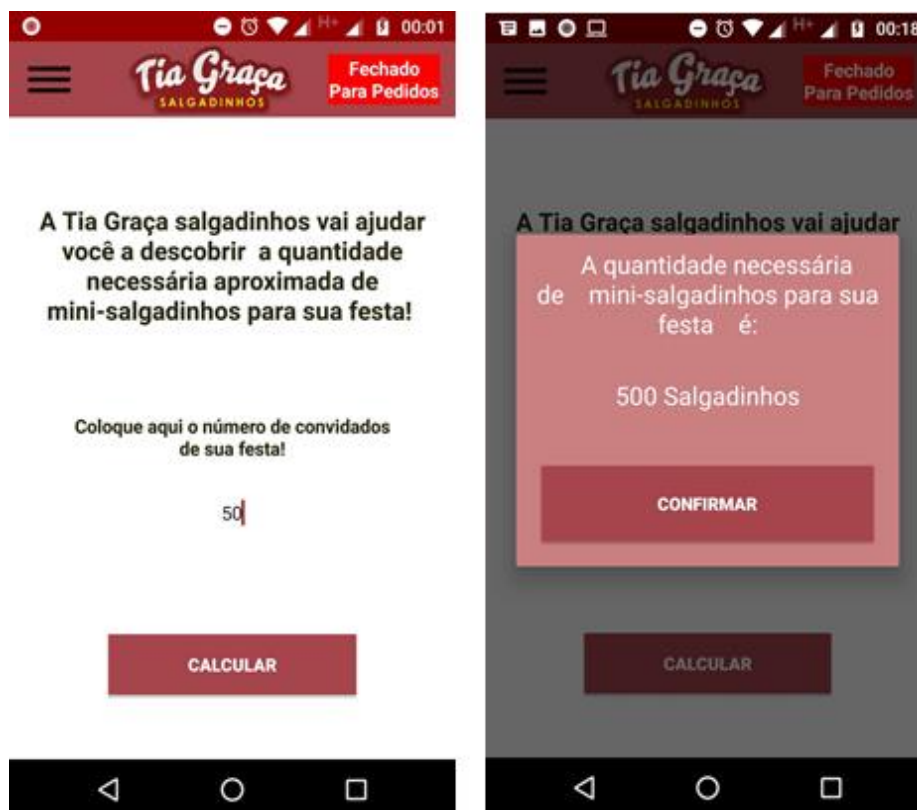
4.5.4 Tela de cálculo

Na tela da Figura 19 apresenta-se um cálculo simples que a dona da empresa ensinou ao grupo de como ela faz as contas para saber a quantidade aproximada de cada encomenda (número de pessoas previsto para a festa x 10 mini-salgadinhos).

Esse resultado é aproximado pois o consumo pode variar dependendo do perfil de cada convidado, pois alguns pode comer mais e outros menos e outro fator que a Tia Graça comentou que a quantidade de adultos e criança também pode diferenciar o resultado.

É um cálculo simples, mas é uma informação importante para o usuário, porque facilita a ele saber aproximadamente quantos salgadinhos terá que agendar, depois de feito o cálculo o usuário é direcionado para a tela de seleção de mini-salgadinhos, a qual será a próxima tela a ser analisada.

Figura 19 - Tela calcular App Tia Graça Salgadinho



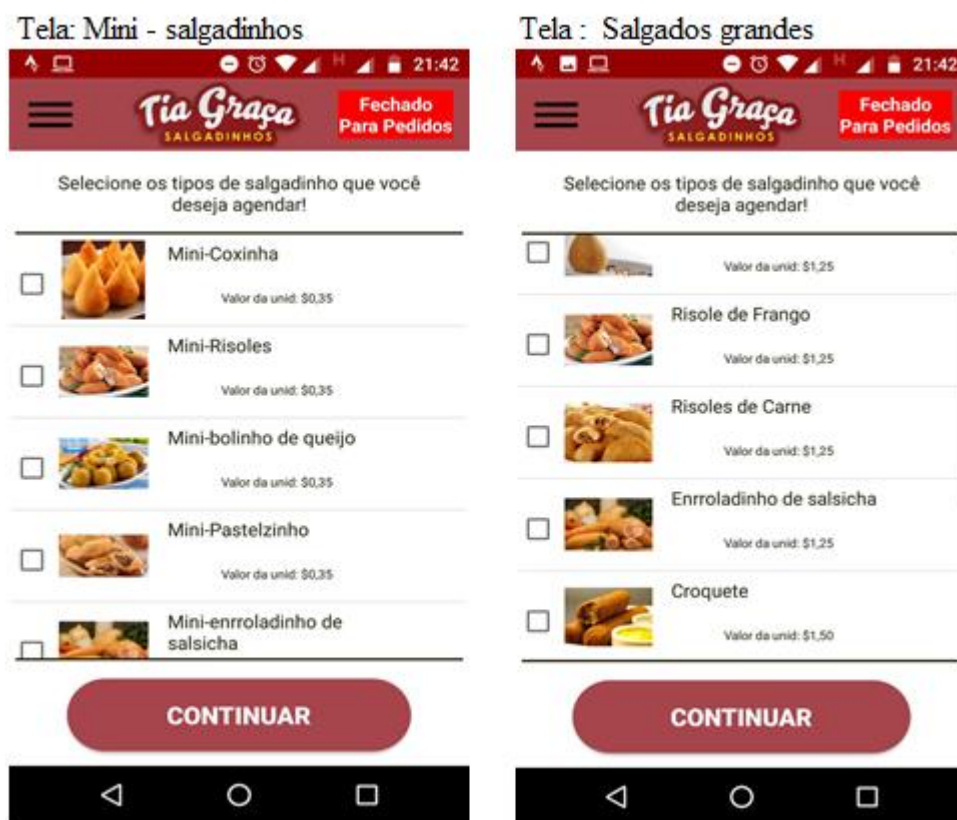
Fonte: Desenvolvida pelos autores (2018).

4.5.5 Tela de seleção de itens

Nas telas de seleção de itens (Figura 20), apresenta-se os tipos de produtos que a empresa trabalha em uma lista para que o usuário selecione os tipos de salgados ou de mini-salgadinhos. Durante a seleção o usuário pode selecionar um ou mais produtos.

As outras informações adicionadas correspondem ao valor de cada item, nome de cada produto e sua imagem para identificar melhor cada item.

Figura 20 - Tela selecionar mini-salgadinhos e tela selecionar salgados grandes do App Tia Graça Salgadinho

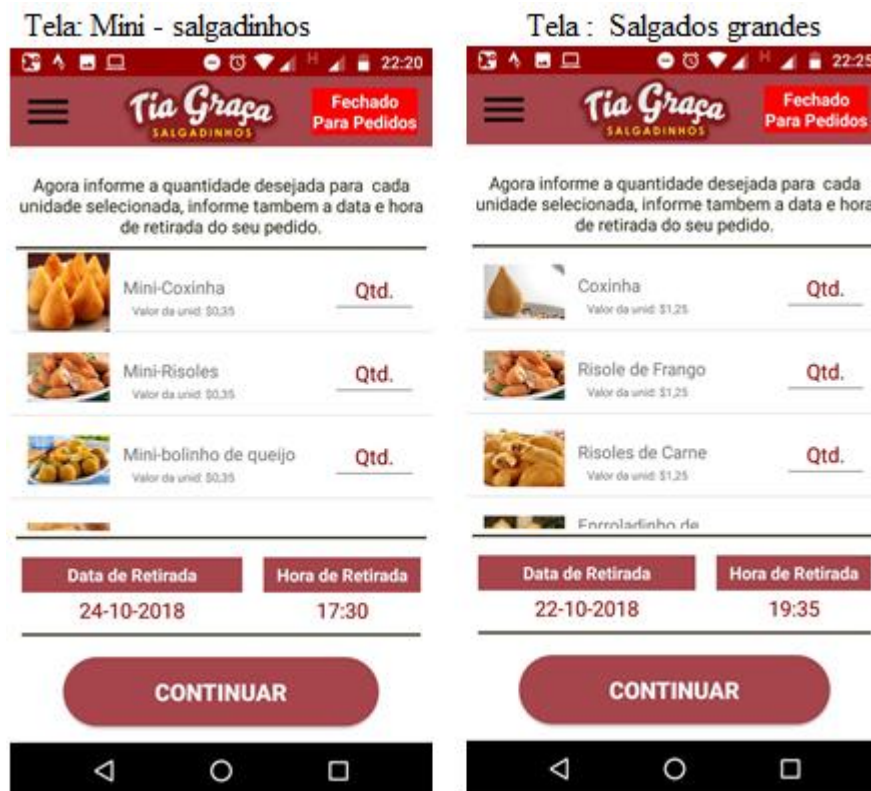


Fonte: Desenvolvida pelos autores (2018).

4.5.6 Tela de quantidades de itens selecionados

Nas telas apresentadas na Figura 21 o usuário tem de colocar as quantidades de cada item selecionado não existe limite de quantidade máxima de produtos, porém a única restrição é que para continuar para próxima tela todos os itens têm que conter pelo menos uma unidade. Depois que colocar as quantidades em cada produto selecionado no próximo passo o usuário terá que informar a data e a hora que pretende retirar seu pedido na Tia Graça salgadinhos. Neste campo o usuário não poderá colocar uma data inferior a 48 horas da data atual.

Figura 21 - Tela quantidade mini-salgadinhos e tela quantidade salgados grandes



Fonte: Desenvolvida pelos autores (2018)

4.5.7 Tela de informações do pedido e formas de pagamento

Na tela da Figura 22, mostra-se todos os dados que foram selecionados nas telas anteriores para que haja uma conferência do usuário, mostra-se também o valor total do pedido feito e a data e hora que o pedido estará pronto para ser retirado, dessa forma o usuário pode conferir os itens selecionados.

Figura 22 - Tela para conferir os dados dos pedidos e informar a forma de pagamento do App Tia Graça Salgadinho

The screenshot displays the 'Tia Graça SALGADINHOS' app interface. At the top, there's a red header with a menu icon, the app logo, and a 'Fechado Para Pedidos' (Closed for Orders) status. Below the header, a message states: 'Você terá que informar o tipo de pagamento para seu pedido.' (You will have to inform the type of payment for your order).

The main content area is divided into sections:

- Produtos** (Products): A table listing three items:

Produto	Valor da unit.	Quantidade	Unidade
Mini-Risoles	\$0,35	30	Unidade
Mini-bolinho de queijo	\$0,35	30	Unidade
Mini-Pastelzinho	\$0,35	30	Unidade
- Total de Mini-Salgadinhos**: 90 Unidades.
- Valor total do pedido**: 31,50 Reais.
- Data da Retirada**: 22-10-2018.
- Hora da Retirada**: 19:30 Horas.

Below these details, a question asks: 'Como você pretende fazer o pagamento na hora de retirar seu pedido?' (How do you intend to make the payment when picking up your order?). The selected option is 'Dinheiro' (Cash). At the bottom, there is a large red button labeled 'FINALIZAR O PEDIDO' (Finish the order).

Fonte: Desenvolvida pelos autores (2018).

4.5.8 Tela de acompanhamento do pedido

A tela de acompanhamento de pedido (Figura 23), foi proposta apenas para informar ao usuário que o pedido que ele confirmou foi enviado para o servidor e em pouco tempo receberá uma confirmação por e-mail, e também depois da confirmação ocorre a mudança do status do pedido no aplicativo.

Figura 23 - Tela acompanha pedido do App Tia Graça Salgadinho



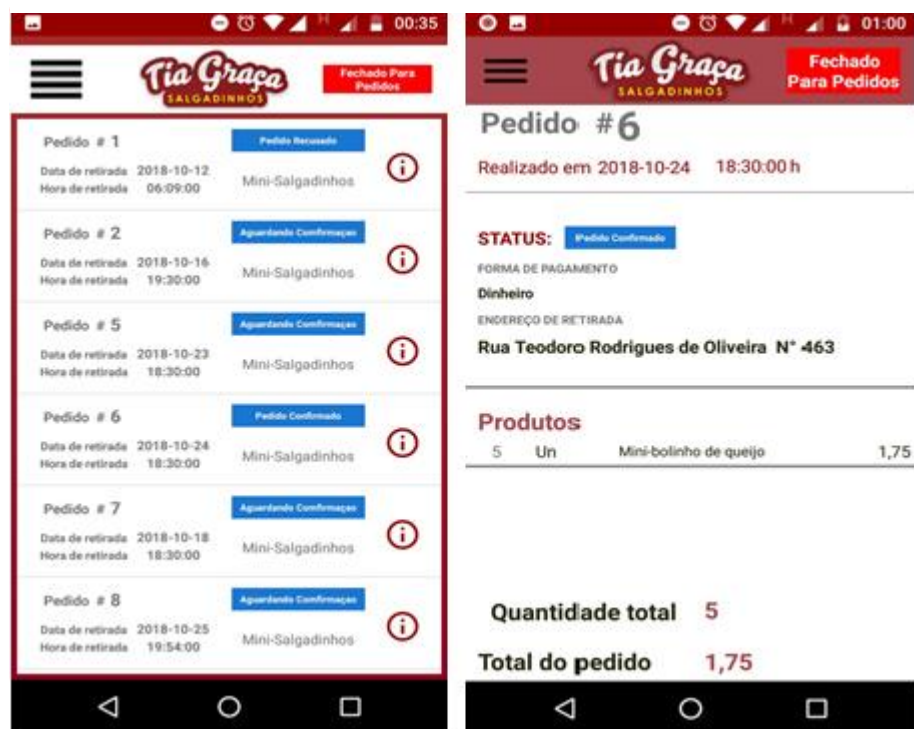
Fonte: Desenvolvida pelos autores (2018).

4.5.9 Tela de acompanhamento do pedido e informação

Na figura 24 são apresentadas as duas telas finais da solução proposta pela equipe, coloca-se todas as informações dos pedidos que o usuário realizou no aplicativo seguido do status do pedido.

Informações como data de retirada, hora, número do pedido e status são as principais informações. Mais informações do pedido podem ser fornecidas a partir da seleção do pedido em questão.

Figura 24 - Tela acompanha pedido do App Tia Graça Salgadinho



Fonte: Desenvolvida pelos autores (2018).

4.6 PROPOSTA DE TELAS PARA CONFIRMAÇÃO DE PEDIDOS

Nesta parte do capítulo, serão mostradas as telas para confirmação de pedidos para desktop e mobile, que foram propostas para a empresa.

4.6.1 Proposta para desktop

Figura 25 - Tela para aceitar pedidos Desktop

====Lista De Pedidos=====

Nome Cliente	Cpf	CódigoPedido	DataRetiradaPedido	HoraRetiradaPedido	TotalDeUnidades	ValorDoPedido	TipoPedido	FormaDePagamento	StatusPedido			
Deivid Jonatan	7894172944	3	2018-10-12	06:16:00	100	35	Mini-Salgadinhos	Cartão de Crédito	Pedido Confirmado	Ver Pedido	Aceitar Pedido	Recusar Pedido
Everton Niotti Favaro	4	10	2018-10-13	06:15:00	157	54.95	Mini-Salgadinhos	Cartão de Débito	Aguardando Confirmação	Ver Pedido	Aceitar Pedido	Recusar Pedido
Everton Niotti Favaro	4	11	2018-10-13	06:34:00	86	30.1	Mini-Salgadinhos	Dinheiro	Aguardando Confirmação	Ver Pedido	Aceitar Pedido	Recusar Pedido
Everton Niotti Favaro	4	17	2018-10-14	06:47:00	100	35	Mini-Salgadinhos	Cartão de Crédito	Aguardando Confirmação	Ver Pedido	Aceitar Pedido	Recusar Pedido
Everton Niotti Favaro	4	21	2018-10-15	06:21:00	100	35	Mini-Salgadinhos	Cartão de Crédito	Aguardando Confirmação	Ver Pedido	Aceitar Pedido	Recusar Pedido
Everton Niotti Favaro	4	20	2018-10-15	09:52:00	100	35	Mini-Salgadinhos	Dinheiro	Aguardando Confirmação	Ver Pedido	Aceitar Pedido	Recusar Pedido
Everton Niotti Favaro	4	2	2018-10-16	19:30:00	158	55.3	Mini-Salgadinhos	Cartão de Crédito	Aguardando Confirmação	Ver Pedido	Aceitar Pedido	Recusar Pedido

Fonte: Desenvolvida pelos autores (2018)

Para a versão *desktop*, foi idealizado inicialmente a proposta de um site em HTML e PHP, que teria funções para aceitar e rejeitar pedidos, então foi feito uma primeira versão simples (Figura 25), e que devido à falta de tempo e o pouco conhecimento dos desenvolvedores sobre a área, foi deixado está proposta para ser desenvolvida em trabalhos futuros e então desenvolvida uma solução com os mesmos objetivos na aplicação desenvolvida em Android.

4.6.2 Proposta para mobile

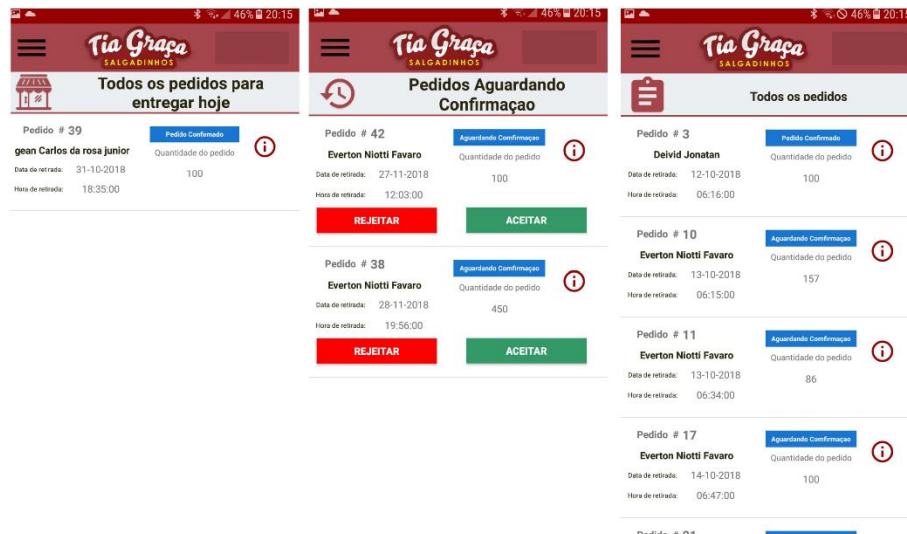
Na Figura 26, pode se observar a tela principal para a parte do administrador do aplicativo, essa tela só poderá ser acessada por um *login* específico, nessa tela o administrador tem 3 opções de ações para serem feitas. Ao ser acessar a opção para ver todos os pedidos para entregar hoje, irá abrir uma tela que exibirá todos os pedidos que tem que ser entregues no dia do acesso (Figura 27, tela 1). Caso seja clicado a opção pedidos aguardando confirmação, irá abrir uma tela onde será exibido todos os pedidos com o status aguardando confirmação e nessa tela o administrador poderá aceitar ou rejeitar o pedido (Figura 27, tela 2). E caso seja clicado a opção para ver todos os pedidos, irá abrir uma tela onde exibe todos os pedidos existentes (Figura 27, tela 3).

Figura 26 - Tela inicial administrador no aplicativo



Fonte: Desenvolvida pelos autores (2018).

Figura 27 - Tela de para exibir pedidos



Fonte: Desenvolvida pelos autores (2018)

5 DESENVOLVIMENTO DO APLICATIVO

Neste capítulo é apresentado o desenvolvimento e implementação da aplicação em Android. A implementação desse aplicativo teve como base um serviço *web*. O aplicativo acessa o Sistema Gerenciador de Banco de Dados (SGBD), por meio de uma ligação cliente-servidor para realizar a ligação entre a plataforma Android e o PHP que é o responsável em acessar o banco de dados e realizar as operações.

5.1 CLIENTE-SERVIDOR

Uma aplicação cliente-servidor tem como base dois componentes, um cliente que requisita um serviço oferecido pelo servidor e um servidor propriamente dito onde estão armazenados os serviços a serem oferecidos. Nesta aplicação a interface cliente-servidor tem como o objetivo promover a interação entre duas plataformas distintas, assim permite que o cliente, neste caso uma aplicação Android, por meio de protocolos HTTP possa solicitar os métodos que estão em arquivos PHP dentro do servidor *web*, e por meio desses métodos realizar as requisições no banco de dados.

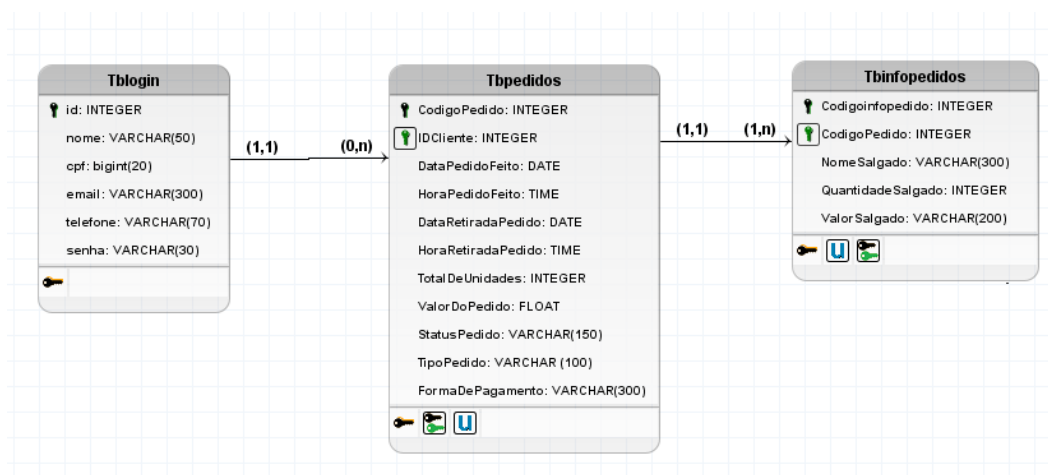
5.2 BANCO DE DADOS

O banco de dados é onde ficará armazenado todos os dados que precisam ser recuperados posteriormente pela aplicação, esses dados são acessados por meio de um SGBD, que consiste de um conjunto de dados, gerenciados por um conjunto de software que permitem a recuperação e o armazenamento desses dados.

No banco de dados ficará salvo todas as informações dos clientes e dos pedidos que foram feitos pelo mesmo. Para realizar o armazenamento dos dados utilizados pela aplicação, foram criadas três tabelas onde ficam armazenados as informações dos clientes e dos seus pedidos como pode ser visto na figura 28. Por meio dessa figura pode-se observar como funciona o relacionamento entre elas, onde cada tabela possui uma chave primária, que é a responsável pela identificação única de cada registro dessa tabela. Algumas tabelas possuem uma chave estrangeira que se refere ao tipo de relacionamento que essa tabela possui, a coluna indicada como chave estrangeira, sempre corresponde a uma coluna que é a chave primária em outra tabela. Esses relacionamentos são necessários para que seja possível realizar, inserções, atualizações e buscas entre essas tabelas. Todas ações de busca, inserções e atualizações nas

tabelas são realizadas por rotinas desenvolvidas em PHP, que estão armazenadas num servidor *web*, e que são acessadas pela aplicação por meio de uma chamada HTTP.

Figura 28 - Modelo lógica do banco de dados



Fonte: Desenvolvido pelos autores (2018)

5.3 A LINGUAGEM PHP

A linguagem PHP é a responsável por receber as requisições HTTP e realizar as operações de inserção, atualização e buscas, sobre as tabelas no banco de dados, ela também é responsável por converter os dados em JSON e enviar de volta ao cliente, neste caso a aplicação Android.

5.4 IMPLEMENTAÇÃO EM PHP

Nesta parte do capítulo, serão apresentados os métodos em PHP que estão no servidor *web* e que realizam a função de servidor da aplicação cliente-servidor. Esses são os métodos que serão empregados pela aplicação para realizar a comunicação com o banco de dados MySQL. Na figura 29, pode-se observar o método responsável pela conexão com o banco de dados.

Figura 29 - Implementação conexão.php

```

1 <?php
2
3 $dsn = "mysql:host=Caminho para o Banco ;dbname=Nome do Banco ;charset=utf8";
4 $usuario = "Usuario";
5 $senha = "Senha";
6
7     try{
8
9         $PDO = new PDO($dsn, $usuario, $senha);
10        //echo "conectou com sucesso";
11    }
12    catch(PDOException $erro ){
13        //echo "Erro: " . $erro->getMessage();
14        echo "conexao_erro";
15        exit;
16    }
17
18 ?>

```

Fonte: Desenvolvido pelos autores (2018)

Na figura 30, pode observar a implementação do arquivo `login.php` que é responsável por receber o e-mail e senha do usuário. Via protocolo HTTP, usando a operação POST e por meio de um método para fazer a comparação com o banco para autorizar o acesso ao aplicativo. O retorno da mensagem de erro ou sucesso na operação de *login* acontece pela criação de uma lista e o retorno dela por meio do método `jsonencode()` e enviadas ao aplicativo pelo método `echo`.

Figura 30 - Implementação do arquivo login.php

```

1 <?php
2     include "conexao.php";
3
4     $email = $_POST['email_app'];
5     $senha = $_POST ['senha_app'];
6
7     $sql_login = "SELECT * FROM Tblogin WHERE email = :EMAIL AND senha = :SENHA";
8     $stmt = $PDO->prepare($sql_login);
9     $stmt->bindParam(':EMAIL', $email);
10    $stmt->bindParam(':SENHA', $senha);
11    $stmt->execute();
12
13    if ($stmt->rowCount() > 0){
14        $retornApp = array("LOGIN=>"SUCESSO");
15
16    } else {
17        $retornApp = array("LOGIN=>"ERRO");
18    }
19
20
21    echo json_encode($retornApp);
22
23 ?>

```

Fonte: Desenvolvido pelos autores (2018)

Nas figuras 31 ,32 e 33, são apresentadas as implementações dos arquivos `login.php` e `cadastro.php`, que também utilizam os métodos HTTP e POST, para o recebimento de informação. Com a diferença de que esses arquivos recebem os dados do aplicativo e salvam

no banco de dados. E, o arquivo `pedidocad.php` também se utiliza dos métodos `str_replace()`, que realiza a substituição de um carácter por outro em uma string. E a `strtotime()`, que é utilizada para converter para o padrão de data que o banco de dados aceita.

Figura 31 - Implementação do Arquivo `cadastrar.php`

```

1 <?php
2     include "conexao.php";
3
4     $nome = $_POST['nome_app'];
5     $cpf = $_POST['cpf_app'];
6     $telefone = $_POST['telefone_app'];
7     $senha = $_POST['senha_app'];
8     $email = $_POST['email_app'];
9
10    $sql_verifica = "SELECT * FROM Tlogin WHERE email = :EMAIL";
11    $stmt = $PDO->prepare($sql_verifica);
12    $stmt->bindParam(':EMAIL', $email);
13    $stmt->execute();
14
15    if($stmt->rowCount() > 0){
16        $retornoApp = array ("CADASTRO" => "EMAIL_ERRO");
17    } else{
18        $sql_insert = "INSERT INTO Tlogin (nome,cpf,senha,telefone,email) VALUES (:NOME, :CPF, :SENHA, :TELEFONE, :EMAIL)";
19        $stmt = $PDO->prepare($sql_insert);
20        $stmt->bindParam(':NOME', $nome);
21        $stmt->bindParam(':CPF', $cpf);
22        $stmt->bindParam(':SENHA', $senha);
23        $stmt->bindParam(':TELEFONE', $telefone);
24        $stmt->bindParam(':EMAIL', $email);
25
26        if($stmt->execute()){
27            $retornoApp = array ("CADASTRO" => "SUCESSO");
28        } else{
29            $retornoApp = array ("CADASTRO" => "ERRO");
30        }
31    }
32
33    echo json_encode($retornoApp);
34
35
36
37
38

```

Fonte: Desenvolvido pelos autores (2018)

Figura 32- Implementação do arquivo `pedidocad.php` parte 1

```

1 <?php
2     include "conexao.php";
3
4     $idCliente = $_POST['id_app'];
5     $DataPedidoFeito = str_replace("/", "-", $_POST['datapedidofeito_app']);
6     $HoraPedidoFeito = $_POST['horapedidofeito_app'];
7     $DataRetirada = str_replace("/", "-", $_POST['dataretirada_app']);
8     $HoraRetirada = $_POST['horaretirada_app'];
9     $TotalDeUnidades = $_POST['totalunidades_app'];
10    $ValorTotalPedido = $_POST['valortotalpedido_app'];
11    $StatusPedido = $_POST['status_app'];
12    $TipoPedido = $_POST['tipo_app'];
13    $FormaDePagamento = $_POST['formadepagamento_app'];
14
15    $DataRetirada = date('Y-m-d', strtotime($DataRetirada));
16
17    $DataPedidoFeito = date('Y-m-d', strtotime($DataPedidoFeito));
18
19    $ValorTotalPedido = str_replace(",", ".", $ValorTotalPedido);
20
21    $sql_verifica = "SELECT * FROM Tlogin WHERE id = :IDCLIENTE";
22    $stmt = $PDO->prepare($sql_verifica);
23    $stmt->bindParam(':IDCLIENTE', $idCliente);
24    $stmt->execute();
25

```

Fonte: Desenvolvido pelos autores (2018)

Figura 33 - Implementação do arquivo pedidocad.php parte 2

```

26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
<?php
    if($stmt->rowCount()<=0){
        $retornoApp =array ("PEDIDO"=>"ID_ERRO");
    } else{
        $sql_insert = "INSERT INTO TbPedidos (IDCliente, DataPedidoFeito, HoraPedidoFeito,
        DataRetiradaPedido, HoraRetiradaPedido, TotalDeUnidades, ValorDoPedido, StatusPedido,
        TipoPedido , FormaDePagamento ) VALUES (:IDCLIENTE, :DATAFEITO, :HORAFEITO, :DATARET, :HORARET,
        :TOTALUNIDADES, :VALORTOTAL, :STATUSPED, :TIPOPEDIDO, :FORMADEPAGAMENTO);";
        $stmt= $PDO->prepare($sql_insert);
        $stmt->bindParam(':IDCLIENTE', $idCliente);
        $stmt->bindParam(':DATAFEITO', $DataPedidoFeito);
        $stmt->bindParam(':HORAFEITO', $HoraPedidoFeito);
        $stmt->bindParam(':DATARET', $DataRetirada);
        $stmt->bindParam(':HORARET', $HoraRetirada);
        $stmt->bindParam(':TOTALUNIDADES', $TotalDeUnidades);
        $stmt->bindParam(':VALORTOTAL', $ValorTotalPedido);
        $stmt->bindParam(':STATUSPED', $StatusPedido);
        $stmt->bindParam(':TIPOPEDIDO', $TipoPedido);
        $stmt->bindParam(':FORMADEPAGAMENTO', $FormaDePagamento);

        if($stmt->execute()){
            $idpedido = $PDO->lastInsertID();
            $retornoApp =array ("PEDIDO"=>"SUCESSO", "ID"=>$idpedido);
        }else{
            $retornoApp =array ("PEDIDO"=>"ERRO");
        }
    }
    echo json_encode($retornoApp);
?>

```

Fonte: Desenvolvido pelos autores (2018)

Nas figuras de 34 a 39, pode-se observar que elas recebem parâmetros também via HTTP e POST, e com esses parâmetros são recuperadas as informações do banco de dados que correspondem a esses parâmetros. Essa recuperação acontece a través de uma execução de um código SQL no banco de dados, assim sendo esses dados são agrupados numa lista e convertidos em JSON pelo método *jsonencode()* e enviado ao aplicativo pelos método *echo()*.

Figura 34 - Implementação do arquivo AcompanharPedido.php

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
<?php
    include "conexao.php";
    $IdCliente = $_POST['idCliente_app'];

    $sql_read = "SELECT CodigoPedido,DataRetiradaPedido,HoraRetiradaPedido,
    StatusPedido,TipoPedido FROM TbPedidos WHERE IDCliente = ".$IdCliente."";
    $stmt = $PDO->query($sql_read);
    $result = array();

    while ($c = $stmt->fetch(PDO::FETCH_OBJ)){
        $result[] = array("CodigoPedido"=>$c->CodigoPedido, "DataRetiradaPedido"=>$c->DataRetiradaPedido,
        "HoraRetiradaPedido"=>$c->HoraRetiradaPedido, "StatusPedido"=>$c->StatusPedido, "TipoPedido"=>$c->TipoPedido);
    }

    echo json_encode($result);
?>

```

Fonte: Desenvolvido pelos autores (2018)

Figura 35 - Implementação do arquivo RecuperarDadosUsuario.php

```

3      $Email = $_POST['email_app'];
4
5      $sql_read = "SELECT nome,email,cpf,telefone FROM Tblogin WHERE email = '". $Email . "'";
6      $stmt = $PDO->query($sql_read);
7      $result = array();
8      while ($c = $stmt->fetch(PDO::FETCH_OBJ)){
9          $result[] = array("nome"=>$c->nome, "email"=>$c->email, "cpf"=>$c->cpf, "telefone"=>$c->telefone);
10     }
11
12     echo json_encode($result);
13 }>

```

Fonte: Desenvolvido pelos autores (2018)

Figura 36 - Implementação do arquivo pedidoinfo.php

```

3
4      $CodigoPedido = $_POST['codigopedido_app'];
5      $NomeSalgado = $_POST['nomesalgado_app'];
6      $QuantidadeSalgado = $_POST['quantidadesalgado_app'];
7      $ValorSalgado = $_POST['valorsalgado_app'];
8
9      $sql_verifica = "SELECT * FROM TbPedidos WHERE CodigoPedido = :CODIGOPEDIDO ";
10     $stmt = $PDO->prepare($sql_verifica);
11     $stmt->bindParam(':CODIGOPEDIDO', $CodigoPedido);
12     $stmt->execute();
13
14     if($stmt->rowCount()<=0){
15         $retornoApp = array ("INFPEDIDO"=>"ID_ERRO");
16     } else{
17         $sql_insert = "INSERT INTO TbPedidosInfo (CodigoPedido, NomeSalgado, QuantidadeSalgado,
18             ValorSalgado ) VALUES (:CODIGOPEDIDO, :NOMESALGADO, :QUANTIDADEDEALGADO, :VALORSALGADO)";
19         $stmt= $PDO->prepare($sql_insert);
20         $stmt->bindParam(':CODIGOPEDIDO', $CodigoPedido);
21         $stmt->bindParam(':NOMESALGADO', $NomeSalgado);
22         $stmt->bindParam(':QUANTIDADEDEALGADO', $QuantidadeSalgado);
23         $stmt->bindParam(':VALORSALGADO', $ValorSalgado);
24         if($stmt->execute()){
25
26             $retornoApp =array ("INFPEDIDO"=>"SUCESSO");
27
28         }else{
29             $retornoApp =array ("INFPEDIDO"=>"ERRO");
30
31         }
32     }
33     echo json_encode($retornoApp);
34 }>

```

Fonte: Desenvolvido pelos autores (2018)

Figura 37 - Implementação do arquivo RetonarDadosPedido.php

```

1 <?php
2     include "conexao.php";
3
4     $CodigoPedido= $_POST['CodigoPedido_app'];
5     $sql_read = "SELECT DataRetiradaPedido,HoraRetiradaPedido,
6         StatusPedido,TipoPedido FROM TbPedidos WHERE CodigoPedido = '". $CodigoPedido . "'";
7     $stmt = $PDO->query($sql_read);
8
9     $result = array();
10    while ($c = $stmt->fetch(PDO::FETCH_OBJ)){
11        $result[] =array( "DataRetiradaPedido"=>$c->DataRetiradaPedido,
12            "HoraRetiradaPedido"=>$c->HoraRetiradaPedido,
13            "StatusPedido"=>$c->StatusPedido, "TipoPedido"=>$c->TipoPedido);
14    }
15    echo json_encode($result);
16 }>

```

Fonte: Desenvolvido pelos autores (2018)

Figura 38 - Implementação do arquivo RetornarInfoPedidos.php

```

1 <?php
2     include "conexao.php";
3
4     $CodigoPedido= $_POST['CodigoPedido_app'];
5
6     $sql_read = "SELECT NomeSalgado,QuantidadeSalgado FROM TbPedidosInfo WHERE CodigoPedido = '". $CodigoPedido. "'";
7     $stmt = $PDO->query($sql_read);
8
9     $result = array();
10    while ($c = $stmt->fetch(PDO::FETCH_OBJ)){
11        $result[] = array( "NomeSalgado"=>$c->NomeSalgado, "QuantidadeSalgado"=>$c->QuantidadeSalgado);
12    }
13    echo json_encode($result);
14 ?>

```

Fonte: Desenvolvido pelos autores (2018)

Figura 39 - Implementação do arquivo recuperarID.php

```

1 <?php
2
3     include "conexao.php";
4     $email = $_POST['email_app'];
5
6     $sql_read = "SELECT id FROM Tblogin WHERE email = '". $email. "'";
7     $stmt = $PDO->query($sql_read);
8
9     while ($c = $stmt->fetch(PDO::FETCH_OBJ)){
10        $result = $c->id;
11        $retornApp = array("RECUPERAR"=>"SUCESSO", "ID"=>$result);
12    };
13    echo json_encode($retornApp);
14 ?>

```

Fonte: Desenvolvido pelos autores (2018)

Nas figuras 40 e 41, podemos observar os códigos em PHP, responsáveis por receber o código do pedido e atualizar seu status no banco de dados, para “Pedido Confirmado” ou “Pedido Recusado”, conforme o arquivo a ser chamado, e então enviar um e-mail para o cliente que esteja relacionado com esse pedido.

Figura 40 - Implementação do arquivo AceitarPedido.php

```

include "conexao.php";
$CodigoPedido= $_POST['CodigoPedido'];
$sql_login = "UPDATE TbPedidos SET StatusPedido = 'Pedido Confirmado' WHERE CodigoPedido = :CODIGO";
$stmt = $PDO->prepare($sql_login);
$stmt->bindParam(':CODIGO', $CodigoPedido);
$stmt->execute();
if ($stmt->rowCount() > 0){
    $retornApp = array("ATT"=>"SUCESSO");
    $sql_email = " SELECT L.nome,L.cpf,L.email,P.StatusPedido,P.DataRetiradaPedido,
P.HoraRetiradaPedido FROM Tblogin AS L JOIN TbPedidos AS P ON L.id =
P.IDCliente WHERE P.CodigoPedido = '". $CodigoPedido. "' ";
    $stmt2 = $PDO->query($sql_email);
    while ($c = $stmt2->fetch(PDO::FETCH_OBJ)){
        $result = $c->cpf;
        ini_set('display_errors', 1);
        error_reporting(E_ALL);
        $from = "teste@tiagracasalgadinhos.com.br";
        $to = $c->email;
        $subject = "Verificando o correio do PHP";
        $message = "Ola,".$c->nome.", O status atual de
seu pedido de numero:". $CodigoPedido. " ,
é: ".$c->StatusPedido.", Ele estará pronto
no dia: ".$c->DataRetiradaPedido." as ".$c->HoraRetiradaPedido." ,
Tia Graça Salgadinhos agradece pela sua preferencia.";
        $headers = "De:". $from;
        mail($to, $subject, $message, $headers);
    };
} else { $retornApp = array("ATT"=>"ERRO"); } echo json_encode($retornApp);

```

Fonte: Desenvolvido pelos autores (2018)

Figura 41 - Implementação do arquivo RecusarPedido.php

```
include "conexao.php";
$CodigoPedido= $_POST['CodigoPedido'];
$sql_login = "UPDATE TbPedidos SET StatusPedido = 'Pedido Recusado' WHERE CodigoPedido = :CODIGO";
$stmt = $PDO->prepare($sql_login);
$stmt->bindParam(':CODIGO', $CodigoPedido);
$stmt->execute();
if ($stmt->rowCount() > 0){
    $retornApp = array("ATT"=>"SUCESSO");
    $sql_email = " SELECT L.nome,L.cpf,L.email,P.StatusPedido,P.DataRetiradaPedido,
P.HoraRetiradaPedido FROM Tblogin AS L JOIN TbPedidos AS P ON L.id = P.IDCliente
WHERE P.CodigoPedido = ".$CodigoPedido." " ;
    $stmt2 = $PDO->query($sql_email);
    while ($c = $stmt2->fetch(PDO::FETCH_OBJ)){
        $result = $c->cpf;
        ini_set('display_errors', 1);
        error_reporting(E_ALL);
        $from = "teste@tiagracasalgadinhos.com.br";
        $to = $c->email;
        $subject = "Verificando o correio do PHP";
        $message = "Ola, ".$c->nome.", O status atual de seu pedido
de numero: ".$CodigoPedido.", é: ".$c->StatusPedido.",
Ele estará pronto no dia: ".$c->DataRetiradaPedido." as "
.$c->HoraRetiradaPedido." ,
Tia Graça Salgadinhos agradece pela sua preferencia.";
        $headers = "De:". $from;
        mail($to, $subject, $message, $headers);
    };
} else {$retornApp = array("ATT"=>"ERRO");}echo json_encode($retornApp);
```

Fonte: Desenvolvido pelos autores (2018)

Nas figuras 42,43 e 44, estão os código em PHP responsáveis por realizar a busca no banco de dado e retornar os dados dos pedidos via JSON para a aplicação, conforme cada implementação específica. Na figura 42, retornando todos os pedidos cuja a situação de status seja igual a “Aguardando Confirmação”, na figura 43 retornando todos os dados de pedidos cujo o status seja “Pedido Confirmado” e a data seja igual ao do dia acessado, e na Figura 44, retorna todos os dados de pedidos que estão no banco de dados.

Figura 42 - Implementação do arquivo PedidosAguardandoConfirmacao.php

```
include "conexao.php";
$sql_read = "SELECT L.nome,P.CodigoPedido,P.DataRetiradaPedido,P.HoraRetiradaPedido,
P.TotalDeUnidades,P.StatusPedido,P.TipoPedido FROM Tblogin AS L JOIN TbPedidos AS
P ON L.id = P.IDCliente WHERE P.DataRetiradaPedido >= CURRENT_DATE AND P.StatusPedido =
'Aguardando Confirmação' ORDER by P.DataRetiradaPedido,P.HoraRetiradaPedido ASC";
$stmt = $PDO->query($sql_read);
$result = array();
while ($c = $stmt->fetch(PDO::FETCH_OBJ)){
    $exploded = explode("-", $c->DataRetiradaPedido);

    $exploded = array_reverse($exploded);

    $c->DataRetiradaPedido = implode("-", $exploded);
    $result[] = array("nome"=>$c->nome, "CodigoPedido"=>$c->CodigoPedido,
    "DataRetiradaPedido"=>$c->DataRetiradaPedido, "HoraRetiradaPedido"=>$c->HoraRetiradaPedido,
    "StatusPedido"=>$c->StatusPedido, "TotalDeUnidades"=>$c->TotalDeUnidades,
    "TipoPedido"=>$c->TipoPedido);
}
echo json_encode($result);
```

Fonte: Desenvolvido pelos autores (2018)

Figura 43 - Implementação do arquivo RetornarTodososPedidoDoDia.php

```
include "conexao.php";

$sql_read = "SELECT L.nome,P.CodigoPedido,P.DataRetiradaPedido,P.HoraRetiradaPedido,
P.TotalDeUnidades,P.StatusPedido,P.TipoPedido FROM Tblogin AS L JOIN TbPedidos AS P
ON L.id = P.IDCliente WHERE P.DataRetiradaPedido = CURRENT_DATE ORDER by
P.DataRetiradaPedido,P.HoraRetiradaPedido ASC";
$stmt = $PDO->query($sql_read);
$result = array();
while ($c = $stmt->fetch(PDO::FETCH_OBJ)){

$exploded = explode("-", $c->DataRetiradaPedido);
$exploded = array_reverse($exploded);

$c->DataRetiradaPedido = implode("-", $exploded);
$result[] = array("nome"=>$c->nome, "CodigoPedido"=>$c->CodigoPedido,
"DataRetiradaPedido"=>$c->DataRetiradaPedido, "HoraRetiradaPedido"=>$c->HoraRetiradaPedido,
"StatusPedido"=>$c->StatusPedido, "TotalDeUnidades"=>$c->TotalDeUnidades, "TipoPedido"=>$c->TipoPedido);
}
echo json_encode($result);
```

Fonte: Desenvolvido pelos autores (2018)

Figura 44 - Implementação do arquivo RetornarTodososPedidos.php

```
include "conexao.php";

$sql_read = "SELECT L.nome,P.CodigoPedido,P.DataRetiradaPedido,P.HoraRetiradaPedido,
P.TotalDeUnidades,P.StatusPedido,P.TipoPedido FROM Tblogin AS L JOIN TbPedidos AS P ON L.id
= P.IDCliente ORDER by P.DataRetiradaPedido,P.HoraRetiradaPedido ASC";
$stmt = $PDO->query($sql_read);

$result = array();

while ($c = $stmt->fetch(PDO::FETCH_OBJ)){

$exploded = explode("-", $c->DataRetiradaPedido);
$exploded = array_reverse($exploded);

$c->DataRetiradaPedido = implode("-", $exploded);
$result[] = array("nome"=>$c->nome, "CodigoPedido"=>$c->CodigoPedido,
"DataRetiradaPedido"=>$c->DataRetiradaPedido, "HoraRetiradaPedido"=>$c->HoraRetiradaPedido,
"StatusPedido"=>$c->StatusPedido, "TotalDeUnidades"=>$c->TotalDeUnidades, "TipoPedido"=>$c->TipoPedido);
}
echo json_encode($result);
```

Fonte: Desenvolvido pelos autores (2018)

5.5 ANDROID STUDIO

Nesta parte descreve-se a parte referente ao Android Studio da aplicação, onde iremos descrever algumas das principais rotinas da aplicação e de como foram implementadas essas funcionalidades.

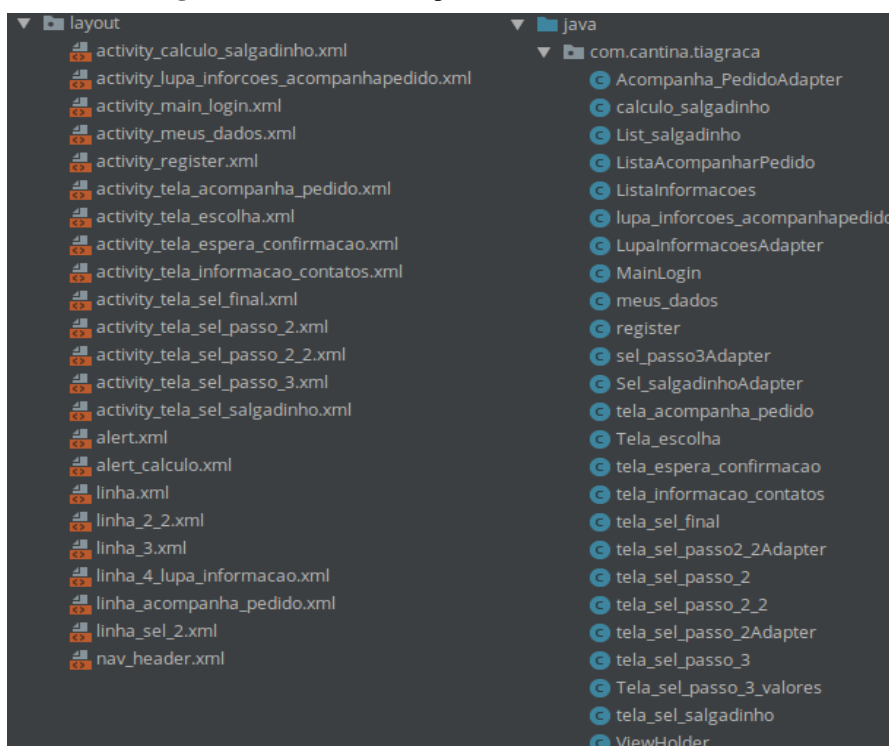
5.5.1 Funcionalidades da aplicação

Esta aplicação foi desenvolvida utilizando a IDE Android Studio, onde ela vem ser a parte principal de todo o projeto, e onde irá acontecer toda a comunicação entre o cliente e a empresa.

A IDE Android Studio, trabalha com o sistema de arquivos .xml que são os responsáveis pela construção da visualização do layout da aplicação e arquivos .java que são responsáveis por conter os métodos para realizar as funcionalidades da aplicação, como ir para outra *activity*,

realizar conexão HTTP com o servidor *web* e muitas outras. Na figura 45 pode-se observar como ocorre a estruturação desses arquivos no ambiente da IDE. Na parte esquerda da figura 45 os arquivos .xml. E na direita da figura 45 observa-se onde ficam os arquivos .java. Nesse arquivo java por meio da funcionalidade *findViewById* é possível recuperar as informações definidas no arquivo XML.

Figura 45 - Estrutura do aplicativo no Android Studio



Fonte: Desenvolvido pelos autores (2018)

Na figura 46, é possível observar o método *logar()*, que é responsável por realizar a chamada HTTP, acessando o arquivo *logar.php* (Figura 30), e passando para ele os parâmetros necessários para realizar o *login* no aplicativo, se esses dados estiverem corretos, será retornado uma mensagem em JSON para o aplicativo, com a mensagem de “SUCESSO”. E então será criado uma *intent* para prosseguir para a *activity* *tela_escolha.xml*, mas caso a mensagem recebida for diferente do esperado, irá ser exibido na tela um *Toast* com uma mensagem de erro.

As classes mencionadas possuem as seguintes funcionalidades: *Intent* – é uma classe android, que representa uma ação que a aplicação quer fazer, desde enviar uma mensagem ao sistema operacional, abrir uma nova *activity* e muitas outras opções. E, *Toast* – é uma classe android, que é utilizado para exibir alertas para o usuário.

Figura 46 - Método `logar` da `MainActivity.java`

```
void logar(final String email, String senha){
    String URL = HOST + "/logar.php";
    if (email.isEmpty() || senha.isEmpty()) {
        Toast.makeText( context: MainLogin.this, text: "Algum campo esta vazio", Toast.LENGTH_LONG).show();
    } else {
        Ion.with( context: MainLogin.this)
            .load(URL)
            .setBodyParameter( name: "email_app", email)
            .setBodyParameter( name: "senha_app", senha)
            .asJsonObject()
            .setCallback((e, result) -> {
                try {
                    String RETORNO = result.get("LOGIN").getAsString();
                    if (RETORNO.equals("ERRO")) {
                        Toast.makeText( context: MainLogin.this, text: "Email ou senha incorreto", Toast.LENGTH_LONG).show();
                    } else if (RETORNO.equals("SUCESSO")) {
                        if (radioButton.isChecked()){
                            preferencia();
                        }
                        Intent abreLogin = new Intent(
                            packageContext: MainLogin.this, Tela_escolha.class);
                        Bundle emailb = new Bundle();
                        emailb.putString("email", email);
                        abreLogin.putExtras(emailb);
                        startActivity(abreLogin);
                    } else {
                        Toast.makeText( context: MainLogin.this, text: "Ocorreu um erro ", Toast.LENGTH_LONG).show();
                    }
                } catch (Exception erro) {
                    Toast.makeText( context: MainLogin.this, text: "Ops,Ocorreu um erro" + erro, Toast.LENGTH_LONG).show();
                }
            });
    }
}
```

Fonte: Desenvolvido pelos autores (2018)

Na figura 47, pode-se observar os métodos `PedidoSalgadinhos()` e `PedidosSalgados()` da aplicação, são eles os responsáveis por carregar os dados que serão apresentados na *listview* dentro de uma lista, e depois carregar essa lista em um *Bundle*, e por meio do método *putSerializable* é configurado as propriedades para que ele possa ser acessado na próxima *activity*, e por meio da função *putExtras* da classe *Intent*, esses dados são passados como parâmetros para a próxima *activity* para que sejam apresentados dentro de uma *listview*. Assim sendo, *Listview* – Desenha os componentes da lista na vertical e com barra de rolagem se necessário.

Figura 47 - Métodos `PedidoSalgadinhos()` e `PedidosSalgados()` da aplicação

```
private void PedidoSalgadinhos(){
    salgadinhos = adicionarsalgadinhos();
    Intent it = new Intent( packageContext: Tela_escolha.this, tela_sel_salgadinho.class);
    Bundle bundle = new Bundle();
    bundle.putSerializable("ListSalgadinhos", salgadinhos);
    it.putExtras(bundle);
    it.putExtras(emailb);
    startActivity(it);
}

private void PedidosSalgados(){
    salgadinhos = adicionarsalgados();
    try{
        Intent it = new Intent( packageContext: Tela_escolha.this, tela_sel_salgadinho.class);
        Bundle bundle = new Bundle();
        bundle.putSerializable("ListSalgadinhos", salgadinhos);
        it.putExtras(bundle);
        it.putExtras(emailb);
        startActivity(it);
    } catch (Exception erro){
        Toast.makeText( context: Tela_escolha.this, text: "Erro: " + erro, Toast.LENGTH_LONG).show();
    }
}
```

Fonte: Desenvolvido pelos autores (2018)

Na figura 48, é mostrado como é feito a recuperação da visualização da lista no arquivo xml, por meio do comando *findViewById*. Depois disso é recuperado os parâmetros que foram passados para essa *activity* por meio do *Bundle*, esses dados recuperados são adicionados a uma lista e então são atribuídos a um *ArrayAdapter* personalizado, e que como pode ser visto na figura 49, em seu método de visualização, este *adapter* já possui as informações de que tipo de dados serão passados e de como serão exibido na tela. Já na figura 50, pode se observar o resultado final dessa junção de .xml com .java, a visualização completa da tela.

Figura 48 - *ListView* preenchimento

```
lista = (ListView) findViewById(R.id.lvlista);
Bundle bundleObject = getIntent().getExtras();
salgadinhos = (ArrayList<List_salgadinho>) bundleObject.getSerializable( key: "ListSalgadinhos");

ArrayAdapter adapter = new Sel_salgadinhoAdapter( context: tela_sel_salgadinho.this, salgadinhos);
lista.setAdapter(adapter);
```

Fonte: Desenvolvido pelos autores (2018)

Figura 49 - *ListView adapter*

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    LayoutInflater inflater = (LayoutInflater) context
        .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

    if(convertView==null) {

        convertView = inflater.inflate(R.layout.linha, parent, attachToRoot: false);
        holder = new ViewHolder();

        holder.nome = (TextView) convertView.findViewById(R.id.nome_mini_salgadinho);
        holder.valor = (TextView) convertView.findViewById(R.id.preco_mini_salgadinho);
        holder.imagem = (ImageView) convertView.findViewById(R.id.imglinha);
        holder.checkBox=(CheckBox) convertView.findViewById(R.id.Ck_tipo_mini);

        holder.checkBox.setOnCheckedChangeListener((tela_sel_salgadinho) context);

        convertView.setTag(holder);
    }
    else {
        holder = (ViewHolder)convertView.getTag();
    }






    List_salgadinho l = elementos.get(position);
    holder.nome.setText(l.getNome());
    holder.valor.setText(l.getValor());
    holder.imagem.setImageResource(l.getImagem());
    holder.checkBox.setTag(l);

    return convertView;
}
```

Fonte: Desenvolvido pelos autores (2018)

Figura 50 - *ListView* visualização

Selecione os tipos de salgadinho que você deseja agendar!

<input type="checkbox"/>		Mini-Coxinha Valor da unid: \$0,35
<input type="checkbox"/>		Mini-Risoles Valor da unid: \$0,35
<input type="checkbox"/>		Mini-bolinho de queijo Valor da unid: \$0,35
<input type="checkbox"/>		Mini-Pastelzinho Valor da unid: \$0,35
<input type="checkbox"/>		Mini-enroladinho de salsicha

Fonte: Desenvolvido pelos autores (2018)

Na figura 51, está o método responsável pela ação de atribuir se o *checkbox* da *listview* está selecionado ou não, cada vez que um *checkbox* é clicado na *ListView*, esta função seta ele como verdadeiro ou falso, assim se ao pressionar o botão para ir para outra tela, apenas os itens da lista que possuem o valor da *checkbox* como verdadeiro, são passados para a próxima tela.

Figura 51- *ListView* visualização

```
@Override
public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
    int pos = lista.getPositionForView(compoundButton);
    if(pos != ListView.INVALID_POSITION){
        List_salgadinho l = salgadinhos.get(pos);
        l.setSelect(b);
    }
}
```

Fonte: Desenvolvido pelos autores (2018)

Na figura 52, está a rotina responsável por pegar os valores que são inseridos no *EditText* da *ListView* (figura 53), e atribuir esses valores a variável quantidade da lista, na posição onde foram informados esses valores. Dentro deste contexto, *EditText* – é uma subclasse da classe *TextView*, e é utilizada para a entrada de informações pelo usuário. *TextView* – uma subclasse da classe *View*, ela representa um texto exibido na tela. *View* – É uma classe utilizada pelo Android para base de todos os componentes gráficos, e também como gerenciados de layout.

Figura 52 - Função pegar dados do *EditText*

```
holder.Quantidade.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
    }

    @Override
    public void afterTextChanged(Editable editable) {
        if(editable != null) {
            elementos.get(position).setQuantidade(editable.toString());
        }
    }
});
```

Fonte: Desenvolvido pelos autores (2018)

Figura 53 - *ListView* com *EditText*

	Mini-Coxinha Valor da unid: \$0,35	20
	Mini-Risoles Valor da unid: \$0,35	Qtd.
	Mini-bolinho de queijo Valor da unid: \$0,35	Qtd.

Fonte: Desenvolvido pelos autores (2018)

Na figura 54, existe a função responsável por acessar todos os dados do pedido que o usuário fez, e por meio de uma chamada HTTP, passar esses dados para os métodos em PHP que estão no servidor *web*, para que estes dados sejam armazenados no banco de dados, depois que isso é feito o servidor *web* envia uma mensagem em JSON, informando que o pedido teve “SUCESSO”, ou uma mensagem de “ERRO”, conforme a mensagem que a aplicação receba ou ela passa o usuário para a próxima tela e então é mostrado uma mensagem ao usuário onde ele deve aguardar, até que seu pedido seja confirmado pela empresa, ou aparece um *Toast* para o usuário que o seu pedido falhou e pede para ele tentar novamente.

Figura 54 - Implementação do cadastro do pedido

```
String URL2 = HOST + "/pedidocad.php";
Ion.with(context: tela_sel_passo_3.this)
    .load(URL2)
    .setBodyParameter( name: "id_app", ID)
    .setBodyParameter( name: "datapedidofeito_app", Data)
    .setBodyParameter( name: "horapedidofeito_app", Hora)
    .setBodyParameter( name: "dataretirada_app", DataRetirada)
    .setBodyParameter( name: "horaretirada_app", HoraRetirada)
    .setBodyParameter( name: "totalunidades_app", TotalUnidade)
    .setBodyParameter( name: "valortotalpedido_app", ValorTotal)
    .setBodyParameter( name: "status_app", StatusPedido)
    .setBodyParameter( name: "tipo_app", TipoPedido)
    .setBodyParameter( name: "formadepagamento_app", FormaPagamento)
    .asJsonObject()
    .setCallback((e, result) => {
        String RETORNO = result.get("PEDIDO").getAsString();

        if (RETORNO.equals("ERRO")) {
            Toast.makeText(context: tela_sel_passo_3.this, text: "Falha ao Realizar o pedido", Toast.LENGTH_LONG).show();
        } else if (RETORNO.equals("SUCESSO")) {
            CodigoPedido = result.get("ID").getAsString();
            for(int i=0; i<salgadinhos.size(); i++){
                String NomeSalgado = salgadinhos.get(i).getNome();
                String QSalgado = salgadinhos.get(i).getQuantidade();
                String ValorSalgado = salgadinhos.get(i).getValor();

                String URL3 = HOST + "/pedidoinfo.php";
                Ion.with(context: tela_sel_passo_3.this)
                    .load(URL3)
                    .setBodyParameter( name: "codigopedido_app", CodigoPedido)
                    .setBodyParameter( name: "nomesalgado_app", NomeSalgado)
                    .setBodyParameter( name: "quantidadesalgado_app", QSalgado)
                    .setBodyParameter( name: "valorsalgado_app", ValorSalgado)
                    .asJsonObject()
                    .setCallback((e, result) => {
```

Fonte: Desenvolvido pelos autores (2018)

Nas figuras 55 e 56, pode-se observar respectivamente as ações responsáveis por chamar os métodos em PHP responsáveis por aceitar e recusar os pedidos sendo eles, AceitarPedido.php (Figura 40) e RecusarPedido.php (Figura 41) que estão hospedados no servidor *web* e só podem ser acessados por meio de uma requisição HTTP.

Figura 55 - Implementação da ação do botão para aceitar pedido

```
holder.Aceitar.setOnClickListener((v) => {
    String URL = HOST + "/AceitarPedido.php";
    Ion.with(v.getContext())
        .load(URL)
        .setBodyParameter( name: "CodigoPedido", elementos.get(position).getNumeroPedido())
        .asJsonObject()
        .setCallback((e, result) => {
            String RETORNO = result.get("ATT").getAsString();

            if (RETORNO.equals("ERRO")) {
                Toast.makeText(v.getContext(), text: "Erro ao aceitar pedido", Toast.LENGTH_LONG).show();
            } else if (RETORNO.equals("SUCESSO")) {
                Toast.makeText(v.getContext(), text: "Pedido Aceito com sucesso", Toast.LENGTH_LONG).show();
                holder.Status.setText("Pedido Confirmado");
                holder.Aceitar.setVisibility(View.INVISIBLE);
                holder.Recusar.setVisibility(View.INVISIBLE);
            }
        });
});
```

Fonte: Desenvolvido pelos autores (2018)

Figura 56 - Implementação da ação do botão para recusar pedido

```
holder.Recusar.setOnClickListener((v) -> {  
    String URL = HOST + "/RecusarPedido.php";  
    Ion.with(v.getContext())  
        .load(URL)  
        .setBodyParameter( name: "CodigoPedido", elementos.get(position).getNumeroPedido())  
        .asJsonObject()  
        .setCallback((e, result) -> {  
            String RETORNO = result.get("ATT").getAsString();  
  
            if (RETORNO.equals("ERRO")) {  
                Toast.makeText(v.getContext(), text: "Erro ao rejeitar pedido", Toast.LENGTH_LONG).show();  
            } else if (RETORNO.equals("SUCESSO")) {  
                Toast.makeText(v.getContext(), text: "Pedido rejeitar com sucesso", Toast.LENGTH_LONG).show();  
                holder.Status.setText("Pedido Recusado");  
                holder.Aceitar.setVisibility(View.INVISIBLE);  
                holder.Recusar.setVisibility(View.INVISIBLE);  
            }  
        });  
});
```

Fonte: Desenvolvido pelos autores (2018)

6 TESTE DO APLICATIVO DESENVOLVIDO NO AMBIENTE DE TRABALHO

Para uma simples avaliação do aplicativo, foram elaboradas três perguntas descritivas e realizado uma entrevista com clientes e funcionários da empresa durante a solicitação de pedidos na Tia Graça Salgadinhos.

Para responder essas questões foi selecionado um grupo de 7 clientes que compram na Tia Graça salgadinhos regularmente, e também foram utilizadas as mesmas questões com os funcionários da empresa.

Todos os clientes selecionados já haviam realizado seus pedidos anteriormente por smartphone, por meio de uma ligação telefônica ou entrando em contato com a empresa pelo seu WhatsApp.

Desta forma foi solicitado aos clientes e aos funcionários a utilização do aplicativo da Tia Graça salgadinhos desenvolvidos pela equipe. A partir disso, foram processadas as respostas para as três perguntas realizadas.

6.1 PERGUNTAS DISSERTATIVAS APRESENTADAS PARA OS USUÁRIOS

A equipe elaborou três perguntas para a entrevista as quais são descritas a seguir:

- **Questão 1: Benefícios ou vantagens de utilizar o aplicativo?**
- **Questão 2: Dificuldades ou desvantagens de utilizar aplicativo?**
- **Questão 3: Quais são as suas sugestões de melhoria para o aplicativo?**

6.2 ANÁLISES DAS RESPOSTAS DOS CLIENTES

A partir das respostas coletadas nas entrevistas com os clientes e funcionários são apresentadas a seguir as principais características citadas nas repostas do grupo de 7 clientes regulares:

- **Questão 1: Benefícios ou vantagens de utilizar o aplicativo?**

Os principais benefícios relatados foram a praticidade na hora de fazer o pedido. Os clientes citaram que não é necessário se locomover até o local e que o pedido pode ser feito

pela internet. Além disso, os clientes citaram que podem realizar o pedido em qualquer horário do dia e que gostaram do design e da organização do aplicativo.

- **Questão 2: Dificuldades ou desvantagens de utilizar aplicativo?**

A maioria dos entrevistados relataram que não tiveram nenhuma dificuldade em utilizar o aplicativo e comentaram que é simples de utilizar. Apenas um entrevistado comentou que encontrou algumas dificuldades em entender os passos necessários para realizar a seleção dos salgadinhos.

- **Questão 3: Quais são as suas sugestões de melhoria para o aplicativo?**

Um único participante apresentou sugestão de melhoria relacionada a adição de um campo que indique o cento de salgadinhos montados em forma de kits. No geral todos os usuários relataram ser muito simples de usar.

6.3 ANÁLISE DAS RESPOSTAS DOS FUNCIONÁRIOS DA EMPRESA

A seguir serão apresentadas as principais características citadas nas respostas dos funcionários.

- **Questão 1: Benefícios ou vantagens de utilizar o aplicativo?**

No relato da empresária os benefícios são bem significativos. A empresária citou que as principais vantagens são a rapidez na hora de fazer os pedidos e o fato de não precisar utilizar o caderno de anotações para registrar os pedidos. Além disso, a empresária citou que é mais fácil de usar comparado com os pedidos realizados por meio do WhatsApp.

A funcionária relatou a praticidade em fazer o pedido e o fato de ter o conhecimento do preço já rapidamente na palma da mão.

- **Questão 2: Dificuldades ou desvantagens de utilizar aplicativo?**

Tanto a empresária e a funcionária relataram que o aplicativo é simples de usar e só precisam se acostumar com a utilização do aplicativo no dia a dia de trabalho.

- **Questão 3: Quais são as suas sugestões de melhoria para o aplicativo?**

A empresária e a funcionária não apresentaram nenhuma sugestão.

6.4 COMENTÁRIOS FINAIS

Observando os resultados dos relatos dos sete clientes concluímos que o aplicativo obteve uma aceitação inicial satisfatória para o agendamento de pedidos de salgadinhos

Todos os participantes receberam muito bem a ideia de fazer o pedido de salgadinhos pelo aplicativo e a maioria não apresentou dificuldades para realizar os pedidos utilizando a tecnologia móvel.

Para a empresária o aplicativo irá tornar o modo de agendar seus pedidos mais produtivo e atenderá as solicitações de seus clientes de uma forma mais eficaz, assim como facilitará o registro manual atual do controle de pedidos.

7 CONCLUSÃO

No desenvolvimento desse trabalho, tivemos a oportunidade de colocar em prática diversos conhecimentos que aprendemos durante o decorrer do nosso curso de graduação. Este trabalho de conclusão de curso nos proporcionou a aplicação de pesquisa e desenvolvimento de sistemas, e também tivemos contatos com tecnologias móveis as quais eram desconhecidas no início da nossa caminhada para conclusão do curso de tecnologia da informação e comunicação.

Os estudos das tecnologias móveis no desenvolvimento do projeto do aplicativo resultou em uma demanda de trabalho maior para o grupo, o qual buscou identificar os possíveis problemas. A solução para os problemas encontrados nas etapas de desenvolvimento foram encontradas por meio de buscas na literatura e estudos de novas plataformas de desenvolvimento móvel.

O projeto buscou a inovação do mercado na região, respeitando a necessidade da empresa em ter um meio de comunicação inovador com os clientes inovador, buscando atender as necessidades da empresa.

A partir das análises dos clientes observamos que o objetivo principal do trabalho foi atingindo, fornecendo a empresa uma mobilidade e uma ferramenta que permite um controle maior de seus pedidos facilitando o seu serviço diariamente.

7.1 TRABALHOS FUTUROS

O aplicativo desenvolvido no escopo deste trabalho será implantado na empresa Tia Graça Salgadinhos no início de 2019.

Como trabalhos futuros, inicialmente iremos procurar melhorar a interface da versão *web* e adicionar relatórios para melhor administração da empresa, tais como, relatórios de total de vendas (organizados por semana, mês e ano) e relatórios das maiores vendas incluindo os principais salgadinhos vendidos. Com esses relatórios espera-se ter uma noção mais abrangente do que produzir e quantidade de vendas realizadas por um determinado período do ano.

Em termos de futuras funcionalidades para o aplicativo pretende-se integrar com outras APIs como o Google Maps para facilitar a identificação da localização da empresa. Também, tem-se como objetivo de trabalhos futuros a realização de testes de usabilidade para identificar possíveis problemas durante a interação com os usuários. E, por fim pretende-se colocar o aplicativo na loja de aplicativos da Google.

8 REFERÊNCIAS

ABRASEL. **Bares e restaurantes devem crescer 4,5% no Brasil. 2017.** Disponível em: <<http://www.abrasel.com.br/component/content/article/7-noticias/5766-21122017-bares-e-restaurantes-devem-crescer-45-no-brasil-neste-ano-setor-estima-aumento-de-25-nas-vendas-apos-ter-sido-afetado-pela-cri-se-economica-mudancas-na-legislacao-trabalhista-ja-comecou-a-beneficiar-a-atividade-no-pais-fortemente-afet.html>>. Acesso em: 20 out. 2018.

ACOM SISTEMAS (Pr - Curitiba) (Org.). **FOOD SERVICE - ALIMENTAÇÃO FORA DO LAR.** 2014. Disponível em: <<http://www.acomsistemas.com.br/blog/food-service-alimentacao-fora-lar/>>. Acesso em: 21 fev. 2018.

ALFALABS. **Aplicativos AlfaLabs.** 2018. Disponível em: <<http://www.alfalabs.com.br/>>. Acesso em: 28 set. 2018.

ALLAN. **Artigo Web Services em aplicações Android e iOS.** 2014. Disponível em: <<https://www.devmedia.com.br/web-services-em-aplicacoes-android-e-ios/28901>>. Acesso em: 01 out. 2018.

ALMEIDA, Leonardo. **O comércio online B2B de alimentos e bebidas.** 2017. Disponível em: <<https://www.ecommercebrasil.com.br/artigos/b2b-alimentos-bebidas/>>. Acesso em: 04 nov. 2018.

ANDROID INC (Brasil). **Conheça o Android Studio.** 2018. Disponível em: <<https://developer.android.com/studio/intro/?hl=pt-br>>. Acesso em: 10 jun. 2018.

APPTEAM. **Burger king.** 2018. Disponível em: <https://play.google.com/store/apps/details?id=com.citrusbits.burgerking&hl=pt_BR>. Acesso em: 02 out. 2018.

BERTÃO, Naiara. **Com apetite de unicórnio, iFood quer se reinventar.** 2018. Disponível em: <<https://exame.abril.com.br/revista-exame/com-apetite-de-unicornio/>>. Acesso em: 09 set. 2018.

BOTTENTUIT JUNIOR, João Batista; PEDROSO, Luciane Freitas. **COMÉRCIO E MARKETING NA WEB: um estudo sobre os sites mais populares de vendas coletivas.** Disponível em: <<http://revistasapientia.inf.br/edicao3/arquivos/2011/BOTTENTUIT.pdf>>. Acesso em: 18 set. 2012.

BURGER King. 2006. Disponível em: <<http://mundodasmarcas.blogspot.com/2006/05/burger-king-land-of-whopper.html>>. Acesso em: 28 ago. 2018.

CAPITANIO, Diego. **PROPOSTA DE WEB SITE PARA A PRÁTICA DE DELIVERY DE FAST FOOD: O CASO POLLY DOG LANCHES**. 2008. 103 f. TCC (Graduação) - Curso de Curso de Administração e Análise de Sistemas, Centro Universitário Univantes, Lajeado, 2008.

CONTENT, Abril Branded. **7 tendências para o mercado de food service**. 2017. Disponível em: <<https://exame.abril.com.br/pme/7-tendencias-para-o-mercado-de-food-service/>>. Acesso em: 01 mar. 2018.

CRONIN, M. **The Internet as a Competitive Business Resource**. In CRONIN, M. **The Internet Strategy Handbook**. Boston: Harvard Business School Press, 1996.

E-COMMERCE NEWS (Brasil). **O que é E-Commerce?** 2017. Disponível em: <<https://ecommercenews.com.br/o-que-e-e-commerce/>>. Acesso em: 8 mar. 2018.

EDUARDO. **O que é HTML 5**. 2012. Disponível em: <<https://www.devmedia.com.br/o-que-e-o-html5/25820>>. Acesso em: 24 set. 2018.

FERNANDES, Jose. **Tipos de comércio eletrônico**. 2012. Disponível em: <<https://bloomidea.com/blog/tipos-comercio-electronico>>. Acesso em: 19 jul. 2012.

FILGUEIRAS, Fellipe. **Java: A origem**. 2015. Disponível em: <<https://tableless.com.br/java-origem/>>. Acesso em: 23 abr. 2018.

FONTANA JUNIOR, Sergio Antonio. **Protótipo de um aplicativo Android para pedidos de lanches e um portal web para gestão e monitoramento**. 2013. 131 f. Monografia (Especialização) - Curso de Sistemas de Informação, Universidade do Planalto Catarinense, São Paulo, 2013.

FRANCO JR., Carlos F. **e-Business: tecnologia de informação e negóciosna internet**. 2.ed. São Paulo: Atlas, 2003.

FREDERICK, Gail Rahn; LAL, Rajesh. **Dominando o Desenvolvimento Web Para Smartphone**. Rio de Janeiro: Alta Books, 2011. 360 p.

G1. **Pedidos de comida por aplicativo crescem em Campinas, diz pesquisa**. 2016. Disponível em: <<http://g1.globo.com/sp/campinas-regiao/noticia/2016/04/pedidos-de-comida-por-aplicativo-cresce-em-campinas-diz-pesquisa.html>>. Acesso em: 10 maio 2018.

G1. **Brasil tem 116 milhões de pessoas conectadas à internet, diz IBGE.** 2018. Disponível em: <<https://g1.globo.com/economia/tecnologia/noticia/brasil-tem-116-milhoes-de-pessoas-conectadas-a-internet-diz-ibge.ghtml>>. Acesso em: 09 maio 2018.

IFB. Instituto Foodservice Brasil. **Atuação do IFB.** 2018. Disponível em: <<http://www.institutofoodservicebrasil.org.br/>>. Acesso em: 13 ago. 2018.

IFOOD. **Ifood Site.** Disponível em: <<https://ifood.delivery/br/sobre>>. Acesso em: 18 maio 2018.

LECHETA, Ricardo R. **Google Android:** Aprender a criar aplicações para dispositivos móveis com Android SDK. 3. ed. São Paulo: Novatec, 2013. (ISBN:978-85-7522-344-4).

LUCENA, Francisco. **Google lança ferramenta de desenvolvimento Android Studio para atrair novos devs.** Disponível em: <<https://www.tudocelular.com/android/noticias/n46869/google-lanca-ferramenta-de-desenvolvimento-android.html>>. Acesso em: 29 set. 2018.

MACHADO, Gabriela. **Alimentação fora do lar: tendências e implicações.** Disponível em: <<http://foodsafety.myleus.com/alimentacao-fora-do-lar-tendencias-e-implicacoes/>>. Acesso em: 02 set. 2017.

MEYER, Maximiliano. **A história do Android [Atualizado Android Oreo 8.1].** 2017. Disponível em: <<https://www.oficinadanet.com.br/post/13939-a-historia-do-android>>. Acesso em: 26 jan. 2018.

MOZILLA E COLABORADORES INDIVIDUAIS. **JavaScript.** 2018. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Aprender/JavaScript>>. Acesso em: 11 out. 2018.

NIEDERAUDER, Juliano. **Desenvolvendo Websites com PHP:** Aprenda a criar Websites dinâmicos e interativos com PHP e banco de dados. São Paulo: Novatec Editora, 2004. 296 p.

PEREIRA, Adalberto Pinto; PRADO, André Alves. **Comércio eletrônico: vantagens competitivas para empresas no B2C (Empresa-para-Consumidor).** 2016. Disponível em: <<http://publicacoes.fatea.br/index.php/raf/article/viewFile/220/177>>. Acesso em: 15 maio 2017.

PRODANOV, Cleber Cristiano e FREITAS, Ermani César de. **Metodologia do trabalho científico [recurso eletrônico] métodos e técnicas da pesquisa e do trabalho científico.** 2ª ed. Novo Hamburgo Feevale, 2013.

RAMEZ ELMASRI SHAMKANT B. NAVATHE (São Paulo). **Sistema de Banco de Dados**. 4. ed. São Paulo: Pearson Education do Brasil, 2005. 7 p.

SAMPAIO, Daniel. **Marketplace não é e-commerce!:** Descubra a diferença e como conseguir resultados com essa ferramenta. Disponível em: <<https://marketingdeconteudo.com/marketplace/>>. Acesso em: 08 out. 2018.

SCUADRA (Brasil). **Conheça 4 tendências de Food Service para 2018**. 2018. Disponível em: <<https://www.scuadra.com.br/blog/conheca-4-tendencias-de-food-service-para-2018/>>. Acesso em: 10 nov. 2018.

SILVA, Henrique Manduca da. **COMUNICAÇÃO ENTRE PLATAFORMAS PHP E JAVA ANDROID: DESENVOLVENDO UM SISTEMA PARA AGENDAMENTO ONLINE DE SERVIÇOS**. 2014. 62 f. Monografia (Especialização) - Curso de Licenciatura em Informática Desenvolvimento de Sistemas Para Internet e Dispositivos Móveis, Universidade Tecnológica Federal do Paraná, Francisco Beltrão-pr, 2014.

SILVA, Vinícius da; TOFFOLO, Geferson. **DESENVOLVIMENTO DE UM SISTEMA WEB PARA GERENCIAMENTO DA ACADEMIA POWER LIFE DE PALMITINHO/RS**. 2016. 20 f. TCC (Graduação) - Curso de Gestão da Tecnologia da Informação, Fai Faculdades, Aaa, 2016.

SOARES, Rodrigo Oliveira; HOPPEN, Norberto. **Aspectos do Uso da Internet nos Negócios pelas Grandes Empresas no Brasil: Um Estudo Exploratório Baseado em Sites Web**. 1997. Disponível em: <<http://www.anpad.org.br/admin/pdf/enanpad1998-ai-06.pdf>>. Acesso em: 02 nov. 2017.

TABLELESS. **O que é CSS?** Disponível em: <<http://tableless.github.io/iniciantes/manual/css/>>. Acesso em: 09 jun. 2018.

TEIXEIRA, Tarcísio. **Comercio Eletrônico - Conforme o Marco Civil Da Internet e a regulamentação do e-commerce no Brasil**. São Paulo: Saraiva, 2015.

TERRA, Portal (Org.). **Como os aplicativos de delivery vêm mudando o mercado de entrega de comida no Brasil**. 2017. Disponível em: <<https://www.terra.com.br/noticias/dino/como-os-aplicativos-de-delivery-vem-mudando-o-mercado-de-entrega-de-comida-no-brasil,84aca079c0cca3e5427dde85eac051c5jb5ph25h.html>>. Acesso em: 26 fev. 2018.

VINÍCIUS, Thiago. **Java: história e principais conceitos**. Disponível em:

<<https://www.devmedia.com.br/java-historia-e-principais-conceitos/25178>>. Acesso em: 11 fev. 2018.

9 ANEXOS

Anexo A: Cliente 1

Perguntas respondida por: *Cliente 1*

Data: *27/10*

Questão 1: Benefícios ou vantagens de se utilizar o aplicativo?

Praticidade, Rapidez, Não precisa sair de casa para fazer o pedido

Questão 2: Dificuldades ou desvantagens de utilizar o aplicativo?

Nenhuma

Questão 3: Quais são as suas sugestões de melhoria do Aplicativo?

Nenhuma, Achou muito fácil de usar

Anexo B: Cliente 2

Perguntas respondida por: *cliente 2*
Data: *27/10*

Questão 1: Benefícios ou vantagens de se utilizar o aplicativo?

*Praticidade no atendimento.
Bonito, bem organizado*

Questão 2: Dificuldades ou desvantagens de utilizar o aplicativo?

Nenhuma

Questão 3: Quais são as suas sugestões de melhoria do Aplicativo?

*Colocar mais claro o passo a passo na
hora de fazer o pedido.*

Anexo C: cliente 3

Perguntas respondida por: *cliente 3*

Data: *27/10*

Questão 1: Benefícios ou vantagens de se utilizar o aplicativo?

Prático, podendo fazer o pedido utilizando a internet.

Não precisa mais ligar.

Questão 2: Dificuldades ou desvantagens de utilizar o aplicativo?

Nenhuma, Bem simples de usar.

Questão 3: Quais são as suas sugestões de melhoria do Aplicativo?

Nenhuma sugestão.

Anexo D: Cliente 4

Perguntas respondida por: *Cliente 4*
Data: *27/10*

Questão 1: Benefícios ou vantagens de se utilizar o aplicativo?

Agilidade, não precisa sair de casa para fazer o pedido.

Questão 2: Dificuldades ou desvantagens de utilizar o aplicativo?

Nenhuma dificuldade.

Questão 3: Quais são as suas sugestões de melhoria do Aplicativo?

Nenhuma sugestão, Simples de usar

Anexo E: cliente 5

Perguntas respondida por: cliente 5

Data: 27/10

Questão 1: Benefícios ou vantagens de se utilizar o aplicativo?

Praticidade, Comodidade, em qualquer horário
Pode ser feito o pedido.

Questão 2: Dificuldades ou desvantagens de utilizar o aplicativo?

Não selecionar dos salgadinho teve um pouco de
dificuldade, não ~~então~~ entendeu os passo a passo
direito.

Questão 3: Quais são as suas sugestões de melhoria do Aplicativo?

Não, Achei funcional e simples.

Anexo F: Cliente 6

Perguntas respondida por: *cliente 6*

Data: *27/10*

Questão 1: Benefícios ou vantagens de se utilizar o aplicativo?

Rápido na hora de fazer o pedido.

*Pode ser feito o pedido na hora, sem precisar
ligar*

Questão 2: Dificuldades ou desvantagens de utilizar o aplicativo?

Não, Nenhuma

Questão 3: Quais são as suas sugestões de melhoria do Aplicativo?

Nenhuma, Achou muito bom o Aplicativo.

Anexo G: Cliente 7

Perguntas respondida por: *cliente 7*
Data: *27/10*

Questão 1: Benefícios ou vantagens de se utilizar o aplicativo?

*Não precisam ligar para fazer o pedido.
Facil de usar e fazer o pedido.*

Questão 2: Dificuldades ou desvantagens de utilizar o aplicativo?

Nenhuma dificuldade.

Questão 3: Quais são as suas sugestões de melhoria do Aplicativo?

Nenhuma sugestão, está tudo certo para mim

Anexo H: Proprietária

Perguntas respondida por: *Proprietária*
Data: *27/10*

Questão 1: Benefícios ou vantagens de se utilizar o aplicativo?

*Rápido na hora de fazer o pedido
os pedidos estão todos organizados no mesmo local
não precisa mais utilizar a Agenda
não vai mais se esquecer de anotar os pedidos.*

Questão 2: Dificuldades ou desvantagens de utilizar o aplicativo?

Fácil de usar, não tem nenhuma dificuldade.

Questão 3: Quais são as suas sugestões de melhoria do Aplicativo?

*Pode melhorar na hora de selecionar os
salgadinhos, podendo ter um campo de seleção
para todos os salgadinhos.*

Anexo I: Funcionaria

Perguntas respondida por: *Funcionário*

Data: *27/10*

Questão 1: Benefícios ou vantagens de se utilizar o aplicativo?

- não precisa mais fazer uma ligação para fazer o pedido
- Pode fazer o pedido em casa, Não precisa sair de casa para fazer o pedido.
- Bem Simples de usar

Questão 2: Dificuldades ou desvantagens de utilizar o aplicativo?

não, nenhuma dificuldade.

Questão 3: Quais são as suas sugestões de melhoria do Aplicativo?

não tem nenhuma melhoria.